User's Manual

ELPIDA

SYNCHRONOUS DRAM

Document No. E0124N10 (Ver.1.0)
(Previous No. M12394EJ2V2AN00)
Date Published May 2001 CP(K)

Elpida Memory, Inc. is a joint venture DRAM company of NEC Corporation and Hitachi, Ltd.

SUMMARY OF CONTENTS

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

Rambus, RDRAM and the Rambus logo are registered trademarks of Rambus Inc.
Direct Rambus, Direct RDRAM, RIMM and SO-RIMM are trademarks of Rambus Inc.
Pentium is a trademark of Intel Corporation.

The information in this document is subject to change without notice. Before using this document, confirm that this is the latest version.

Elpida Memory, Inc. does not assume any liability for infringement of any intellectual property rights (including but not limited to patents, copyrights, and circuit layout licenses) of Elpida Memory, Inc. or third parties by or arising from the use of the products or information listed in this document. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of Elpida Memory, Inc. or others.

Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of the customer's equipment shall be done under the full responsibility of the customer. Elpida Memory, Inc. assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

[Product applications]
Elpida Memory, Inc. makes every attempt to ensure that its products are of high quality and reliability. However, users are instructed to contact Elpida Memory's sales office before using the product in aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment, medical equipment for life support, or other such application in which especially high quality and reliability is demanded or where its failure or malfunction may directly threaten human life or cause risk of bodily injury.

[Product usage]
Design your application so that the product is used within the ranges and conditions guaranteed by Elpida Memory, Inc., including the maximum ratings, operating supply voltage range, heat radiation characteristics, installation conditions and other related characteristics. Elpida Memory, Inc. bears no responsibility for failure or damage when the product is used beyond the guaranteed ranges and conditions. Even within the guaranteed ranges and conditions, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Elpida Memory, Inc. products does not cause bodily injury, fire or other consequential damage due to the operation of the Elpida Memory, Inc. product.

[Usage environment]
This product is not designed to be resistant to electromagnetic waves or radiation. This product must be used in a non-condensing environment.

If you export the products or technology described in this document that are controlled by the Foreign Exchange and Foreign Trade Law of Japan, you must follow the necessary procedures in accordance with the relevant laws and regulations of Japan. Also, if you export products/technology controlled by U.S. export control regulations, or another country's export control laws or regulations, you must follow the necessary procedures in accordance with such laws or regulations.
If these products/technology are sold, leased, or transferred to a third party, or a third party is granted license to use these products, that third party must be made aware that they are responsible for compliance with the relevant laws and regulations.

M02 01.2

INTRODUCTION

Readers This User's Manual is intended for user engineers who wish to design and develop application systems using synchronous DRAM (SDRAM) to support high-speed bus clock.

Purpose This User's Manual is designed for users to understand the basic concepts related to connection by introducing connection examples between SDRAM and high-speed CPU.

Organization This User's Manual consists of the following subjects.

- SDRAM
- PCI bus
- Designing of SDRAM controller (connection between SDRAM and PCI bus)

Legend

Active low	:	In this User's Manual, active low is described with pin names and signal names postfixed with "#" or "-".
CAS# before RAS#	:	CAS# before RAS# refresh cycle is abbreviated as CBR refresh cycle.

Caution This User's Manual is a preliminary reference. Information contained in this User's Manual is subject to change without notice.

The descriptions concerning PCI bus in this User's Manual show only the concept of operation and not actual operation. Use the descriptions only for reference purpose upon designing.

CONTENTS

LIST OF FIGURES (1/2)

Figure No.	Title	Page

LIST OF FIGURES (2/2)

Figure No.	Title	Page

LIST OF TABLES

CHAPTER 1 SYNCHRONOUS DRAM

This chapter briefly explains major features of synchronous DRAM (SDRAM) focusing on the differences between synchronous DRAM and the conventional DRAM and methods commonly used for controlling SDRAM to design SDRAM controllers.

For the details of control methods of the conventional DRAM and SDRAM, refer to the User's Manual of each product.

1.1 High-speed RAM

1.1.1 Types of high-speed RAM

RAM is roughly divided into two types: DRAM (Dynamic RAM) and SRAM (Static RAM).

The high-speed DRAM generally includes EDO DRAM, SDRAM, and RDRAM®. The next-generation high-speed DRAM includes DDR SDRAM, DirectRDRAM™, SynchLink DRAM, etc.

(1) FPM DRAM (Fast Page Mode DRAM)

FPM DRAM is a DRAM provided with the page mode of higher-speed than that of the conventional DRAM. Although FPM DRAM executes data input/output only once during one cycle of RAS# in random access, it can continuously execute data input/output during one cycle of RAS# in the page mode. In the page mode, the access time of the second data and thereafter becomes faster.

Other types of DRAM such as NB (Nibble Mode) and SC (Static Column Mode) that realize higher-speed using specifications different from those of FPM DRAM also exist. In 1995, however, FPM DRAM represented approximately 90% of all DRAM shipments.

(2) EDO DRAM (Extended Data Out DRAM)

EDO DRAM is a still faster version of FPM DRAM.

If the data output (read cycle) of FPM DRAM is made higher speed, the data output time becomes shorter.

Since EDO DRAM is provided with extended output functions, the data output time does not become short even in higher speed. Thus, wider range for the timing can be allowed in the data output side, and as a result a speed higher than that of FPM DRAM can be realized.

In addition, since EDO DRAM and FPM DRAM are compatible DRAM that have packages with the same pin configuration, EDO DRAM can easily replace FPM DRAM. In the middle of 1996, EDO DRAM represented approximately 50% of all the DRAM shipment.

(3) SDRAM (Synchronous DRAM)

Specifications different from those of the conventional DRAM are used for SDRAM to realize higher-speed operation. EDO DRAM is provided with extended output functions, but it has a problem of operating frequency similarly to FPM DRAM if higher speed is demanded. Generally, EDO DRAM can be synchronized only up to the clock of approximately 75 MHz.

SDRAM is capable of operation at higher operating frequency than EDO DRAM is. SDRAM, however, has package pin configuration, control signal names, and the number of signals different from those of the conventional DRAM since SDRAM performs controls with an interface different from that of the conventional DRAM and the new specifications.

In the first half of 1997, SDRAM occupied approximately 25% of all the DRAM shipment. It is expected that SDRAM will represent more than 50% of DRAM shipments around 1998 and become the most popular high-speed DRAM. Currently, the mainstream of SDRAM synchronizes with the clock of 66 to 100 MHz. SDRAM that synchronizes with the clock of 125 to 143 MHz is also to appear in the future.

(4) RDRAM (Rambus® DRAM)

RDRAM employs the unique specifications proposed by Rambus Inc.

Although RDRAM operates in synchronization with the clock of 300 MHz, adoption of the dual edge system makes it equivalent to synchronization with the clock of 600 MHz. The dual edge system is a system that can perform controls at two points: the rising edge and the falling edge of one clock.

While the operating frequency of RDRAM is currently 300 MHz, DirectRDRAM aims at 400 MHz. DirectRDRAM has the highest operating frequency of all the high-speed DRAM.

DDR SDRAM, DirectRDRAM, and SynchLink DRAM are next-generation high-speed DRAM. The specifications of these next-generation high-speed DRAM are under examination in order to be put in practical use between 1998 and 2000.

Currently, high-speed DRAM are gradually unified to SDRAM, and these next-generation high-speed DRAM are positioned as the successors of SDRAM.

(5) DDR SDRAM (Double Data Rate SDRAM)

DDR SDRAM is a synchronous DRAM that realizes high-speed data transfer while taking over the specifications of SDRAM as much as possible.

Although DDR SDRAM does not have complete compatibility with SDRAM, DDR SDRAM can be used with simple specification changes of the SDRAM controller since DDR SDRAM has basically the same package pin configuration and the control method as SDRAM.

While SDRAM adopts the single-edge system, which performs controls at a single edge of the basic clock, DDR SDRAM adopts the dual-edge system.

As for interface, DDR SDRAM adopts SSTL interface level in order to realize a speed higher than that of SDRAM. DDR SDRAM aims at operating frequency of 100 MHz.

(6) SynchLink DRAM

SynchLink DRAM adopts the dual-edge system, which performs controls at dual edges of the basic clock.

As for interface, SynchLink DRAM adopts a dedicated protocol called SynchLink interface. SynchLink DRAM aims at operating frequency of 400 MHz.

(7) Dual Port Graphics Buffer

Dual Port Graphics Buffer is a memory dedicated for images, and it is configured with two ports: serial port and RAM port.

The serial port performs read-out to display unit (displaying image), and the RAM port mainly performs write-in to an image memory (drawing). Although the RAM port is basically configured with DRAM, the operation mode for drawing, which is not provided to DRAM, is added in order to simplify designing of graphics systems.

(8) SGRAM (Synchronous Graphics RAM)

SGRAM is a synchronous memory dedicated for images. Although SGRAM is basically configured with SDRAM, the operation mode for drawing, which is not provided to SDRAM, is added in order to simplify designing of graphics systems.

(9) DDR SGRAM (Double Data Rate SGRAM)

DDR SGRAM is a next-generation high-speed SGRAM.

While SGRAM adopts the single-edge system, which performs controls at a single edge of the basic clock, DDR SGRAM adopts the dual-edge system.

(10) SSRAM (Synchronous SRAM)

SSRAM is a synchronous high-speed SRAM.

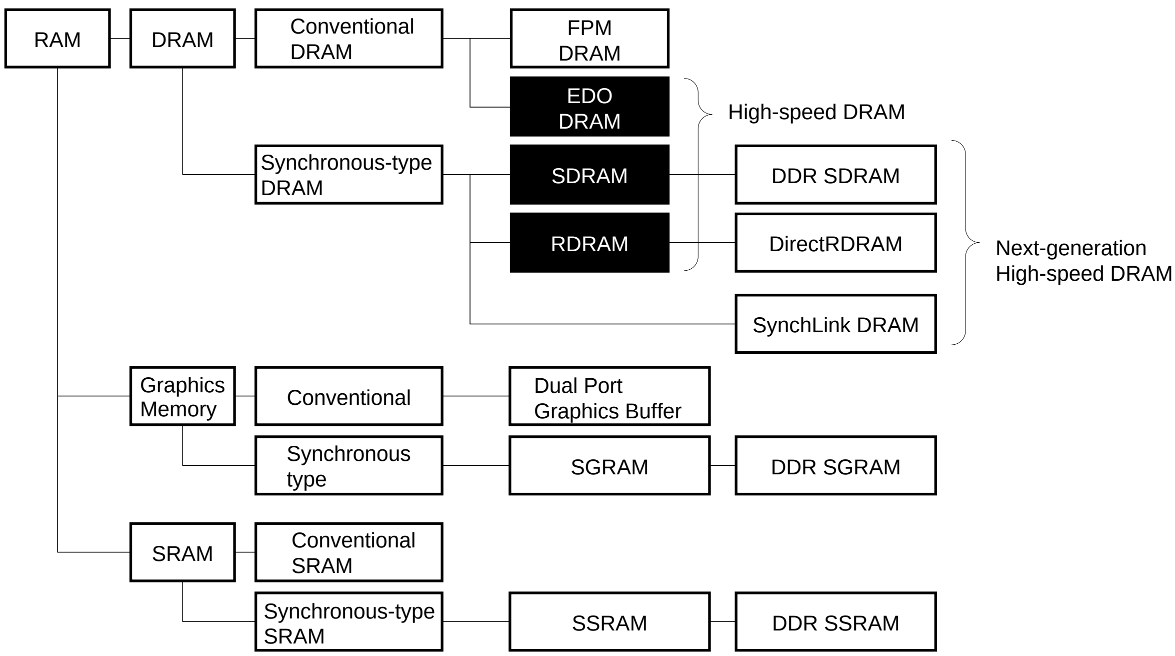
There are two types of SSRAM: the type that adopts the pipeline system and the type that does not adopt the pipeline system (non-pipeline system). Sometimes, the type that adopts the pipeline system is called PBSRAM (Pipeline Burst SRAM), and the type that does not is called SSRAM. (SDRAM adopts the pipeline system).

(11) DDR SSRAM (Double Data Rate SSRAM)

DDR SSRAM is a next-generation high-speed SRAM.

While SSRAM adopts the single-edge system, DDR SSRAM adopts the dual-edge system.

Figure 1-1. Types of High-speed RAM



1.1.2 Access time of high-speed DRAM

Access time of DRAM can be divided into the following two types:

- Random access time : Access time in which both the row address and the column address different from those of the preceding cycle are accessed.
- Burst access time : Access time in which the same row address and a column address different from that of the preceding cycle are accessed.

The DRAM that is newly under development and examination has the random access equal to that of the conventional DRAM, and only its burst access time is made higher speed. As shown in Table 1-1, FPM DRAM, EDO DRAM, SDRAM, and RDRAM have approximately equal random access times but have different burst access times.

Table 1-1. Random Access Time and Burst Access Time

	Random Access Time (ns)	Burst Access Time (ns)
FPM DRAM	50	35
EDO DRAM	50	20
SDRAM	50	8
RDRAM	40	1.7 (1.3) Note
Conventional SRAM	15	–
Synchronous-type SRAM	6	–

Note Figures in the parentheses show the value of DirectRDRAM.

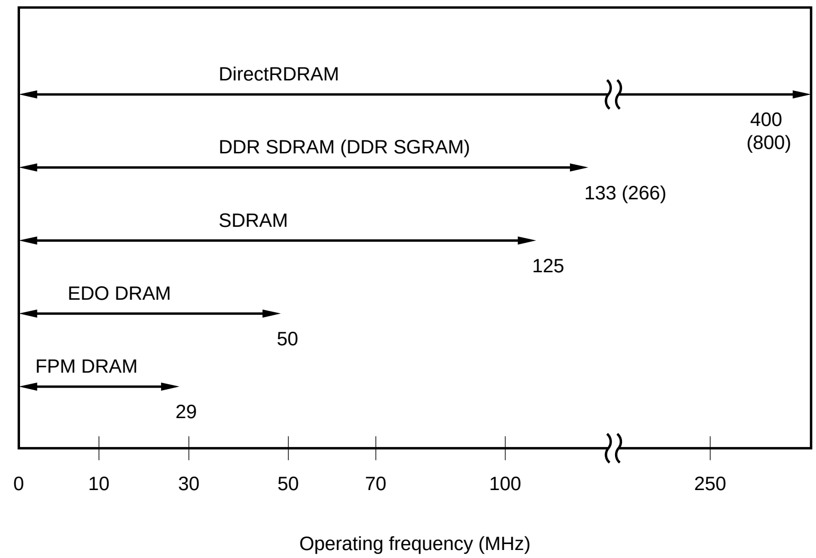
The table above shows that the burst access times of high-speed DRAM are equal to or faster than the random access times of the conventional SRAM and synchronous-type SRAM. Therefore, it is advisable to keep the control with random access minimum and perform controls with burst access as much as possible when controlling high-speed DRAM so that a performance equal or superior to that of the conventional SRAM can be demonstrated.

1.1.3 Relation to system clock

Figure 1-2 shows the random access times and burst access times in Table 1-1 converted into frequencies. The frequencies are likened to memory bus clocks to illustrate up to which memory bus clock each memory can support.

The operating frequencies shown here are conversion of the burst access times of DRAM into operating frequencies. The operating frequencies of the DDR SDRAM (DDR SGRAM) and RDRAM, however, are made half since they adopt the dual-edge system.

Figure 1-2. Memory Bus Clock Each High-speed DRAM Can Support



FPM DRAM had been the main stream until 1996. FPM DRAM supports only a memory bus clock of up to 29 MHz. In order to support higher memory bus clock, controls must be performed inserting waits, so that the performance is equal to the control with the clock of 29 MHz. To realize high-speed access with FPM DRAM, performance of the set was improved implementing complicated controls such as interleaving.

EDO DRAM can improve the performance of a set simply by replacing FPM DRAM.

Neither SDRAM nor RDRAM have compatibility with EDO DRAM since they have different control methods and interfaces. Therefore, when performance better than that of EDO DRAM is needed, the use of higher-speed DRAM such as SDRAM and RDRAM should be considered.

Figure 1-2 shows that the selection of high-speed DRAM is divided at approximately 66 MHz. If FPM DRAM are currently used and memory bus clock of higher than 50 to 66 MHz may not be needed in the future, it is recommended to use EDO RAM, which has the same package pin configuration and interface as those of FPM DRAM, instead of SDRAM, which has different package pin configuration and interface.

If the memory bus clock of 100 MHz or higher will definitely be needed in the future, it is recommended to use SDRAM.

Although the technology of FPM DRAM cannot be inherited, the use of SDRAM will be advantageous in the future because industrial standardization of SDRAM is progressing and the specification to synchronize with the clock of 143 MHz or higher is under examination.

In Figure 1-2, the operating frequency of RDRAM is higher than that of any other DRAM. Although current RDRAM synchronizes with 400-MHz clock, data can be processed at two points: the rising edge and the falling edge of a single clock (dual-edge system), as a result equivalent to synchronization with 800-MHz clock is achieved. If 400-MHz clock is needed for the memory bus clock of a system, the use of RDRAM is effective. In this case, the trunk of a system can be configured by adopting ASIC and DRAM of the Rambus specification since ASIC with the Rambus specification has already been shipped from semiconductor companies.

1.2 Difference between Conventional DRAM and SDRAM

1.2.1 What is SDRAM?

SDRAM is a type of DRAM which operates in synchronization with input clock.

This system has been developed from the idea that the synchronization of the system clock and the operating clock of a memory will make it easier to control each other.

Figure 1-3. Higher-speed SDRAM and EDO DRAM

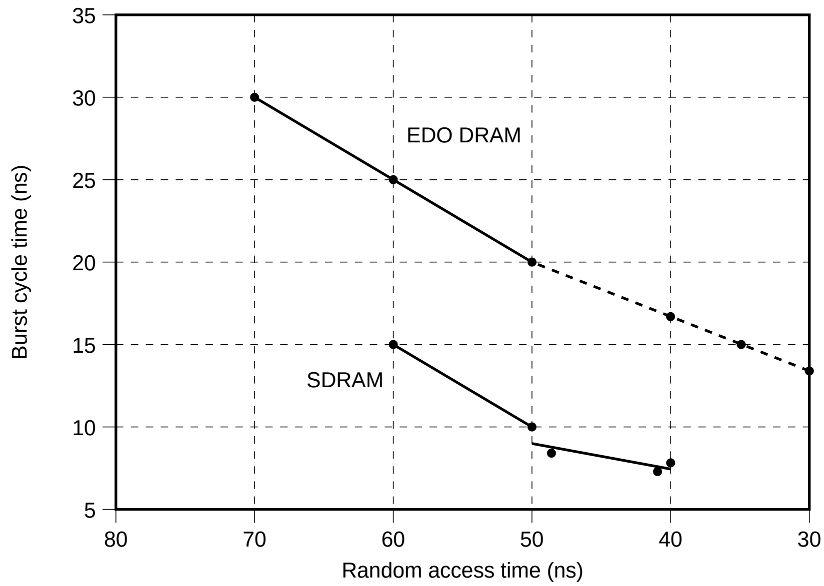


Figure 1-3 shows each access time of SDRAM and EDO DRAM, respectively.

Comparing the access time of SDRAM with that of the conventional DRAM, the burst access time of SDRAM is much faster than that of the conventional DRAM while there is not much difference in the random access time.

Since SDRAM and EDO RAM have almost identical basic configuration inside the memory, there is no difference in basic access times such as random access time. On the other hand, the burst access time of SDRAM is faster than that of the conventional DRAM because SDRAM has adopted technologies such as pipeline system which are different from those of the conventional DRAM.

In Figure 1-3, there are portions shown with the solid line and that shown with the broken line. The portion shown with solid line indicates that the capacity of DRAM is 16 Mbits or more. The portion shown with the broken line indicates that the capacity of DRAM is at most 4 Mbits.

In terms of the burst cycle time, EDO DRAM is capable of synchronization with the clock of 13.3 ns/75 MHz. This value is considered as the limit of EDO DRAM.

It is already planned to make SDRAM synchronize with the clock of 7 ns/143 MHz. It is also under consideration to make SDRAM synchronize with the clock of 200 MHz or higher in the future. The improvement of system performance can be expected in designing a system with fast memory clock of 75 MHz or higher by adopting SDRAM instead of EDO DRAM.

1.2.2 Features of SDRAM

The following explains the features of SDRAM.

(1) Synchronous operation

SDRAM latches each control signal at the rising edge of basic input clock and inputs/outputs data in synchronization with the clock signal. Controls are made easier by synchronizing the clock with the system memory clock.

(2) Controls with commands

A command is a combination of logic levels of control signals.

Typical commands include active command, read or write command, precharge command, etc. The conventional DRAM is also controlled with combinations of logic levels of control signals. The conventional DRAM, however, does not have the concept of command.

(3) Multiple-bank configuration of internal memory circuit

The memory chip is separated into several banks, so that controls can be performed by the bank.

For example, since the interleave control can be performed to each bank, the precharge time is seemingly hidden, thus enabling high-speed access. In the case of SDRAM with 2-bank configuration, the control of the bank is set according to the logic level of the highest address signal.

(4) Adoption of control by the mode register

The mode register sets in advance the CAS# latency and the burst length, etc. in the logic level of address signals at a given time (for the details, refer to **1.3.4 Setting of mode register**). The mode register retains data until the setting is made again or the device loses power.

(5) Selectable CAS# latency (CL)

CAS# latency is the number of clocks from input of a command to output of data. The number of clocks can be set with the mode register. The value of CAS# latency has close relationship with the clock operating frequency. The smaller the value of CAS# latency, the lower the speed of the clock operating frequency must be set to. The larger the value of CAS# latency, the higher the speed of the clock operating frequency can be.

The relationship between CAS# latency and the clock operating frequency differs depending on the product. See each SDRAM Data Sheet in designing a system.

Table 1-2. Example of Clock Operating Frequency Corresponding to CAS# Latency

Grade	–80	–10
CL = 3	125 MHz (8 ns)	100 MHz (10 ns)
CL = 2	87 MHz (12 ns)	66 MHz (15 ns)

Remark CL = 2 cannot be used in the product of grade –80 and –10 at 125 MHz and 100 MHz, respectively.

(6) Selectable burst length (BL)

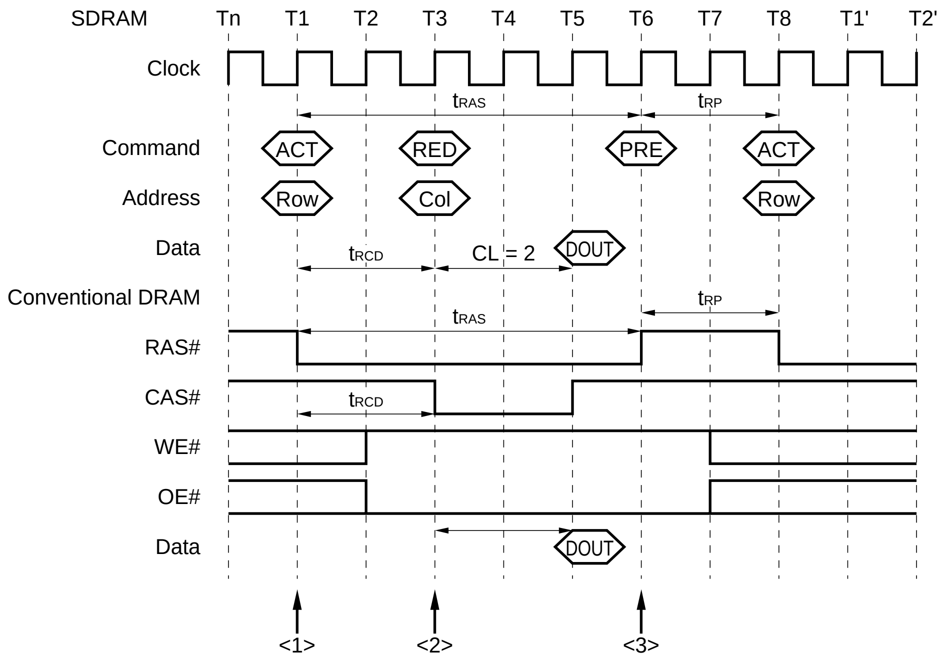
The selection of burst length is made with the mode register. The burst length is a number of words that can continuously be input/output in read cycle or write cycle.

1.2.3 Basic control method and access time

The following explains the actual control method taking the case of read cycle.

(1) Basic control method

Figure 1-4. Read Cycles of Conventional DRAM and SDRAM



At point <1>, active command (ACT) and low address signal are latched at first so that the start of operation is declared.

In the conventional DRAM, this point corresponds to the state that RAS# signal is low.

At point <2>, read command (RED) and column address signal are latched to declare it is a read cycle, and data is output a few clocks after.

In the conventional DRAM, this point corresponds to the state that CAS# signal is made low after RAS# signal becomes low and WE# signal is high.

The time after column address signal is latched until valid data is output is called CAS# latency (CL = 2 in Figure 1-4), and the number of words of the data continuously output is called burst length (BL = 1 in Figure 1-4). In the conventional DRAM, CAS# latency corresponds to CAS# access time, and the burst length corresponds to the number of page mode cycles.

At point <3>, precharge command (PRE) is input t_{RAS} after active command is latched.

In the conventional DRAM, this point corresponds to the state that RAS# signal and CAS# signal are high.

(2) Access time

Figure 1-5. Burst Read Cycle

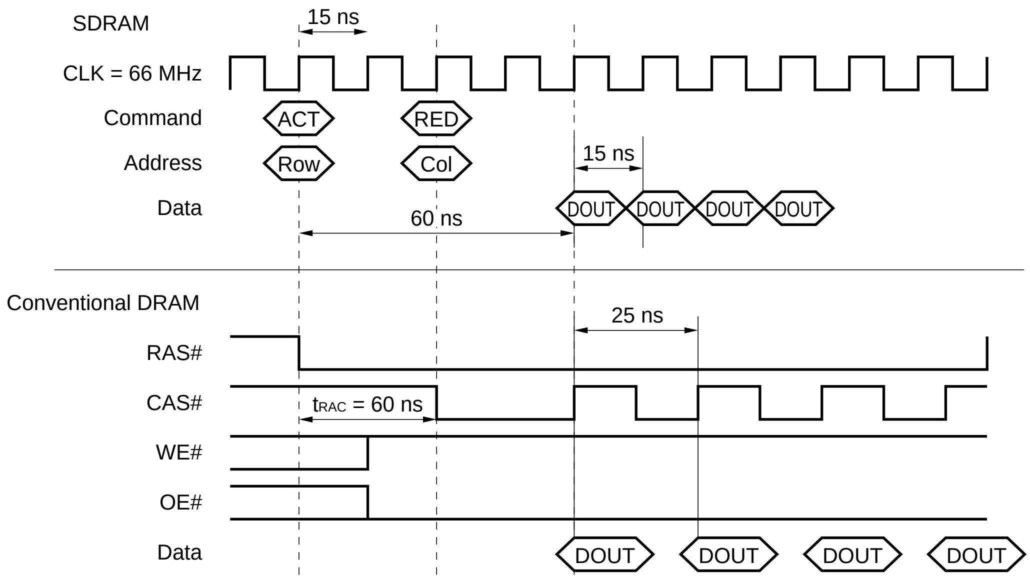


Figure 1-5 shows the burst read cycle when burst length = 4. Assuming the clock speed of SDRAM is 66 MHz, the access time of this SDRAM and that of –60 ns product (t_{RAC}: RAS# access time = 60 ns) of EDO DRAM are compared (the conventional DRAM is not synchronized with 66-MHz clock).

In the case of SDRAM,
1st access = 60 ns
2nd access = 75 ns = 1st access (60 ns) + 15 ns
3rd access = 90 ns = 2nd access (75 ns) + 15 ns
4th access = 105 ns = 3rd access (90 ns) + 15 ns

In the case of EDO DRAM,
1st access = 60 ns
2nd access = 85 ns = 1st access (60 ns) + 25 ns
3rd access = 110 ns = 2nd access (85 ns) + 25 ns
4th access = 135 ns = 3rd access (110 ns) + 25 ns

Comparing SDRAM and EDO DRAM,
SDRAM has the same speed as EDO DRAM does in 1st access but
10 ns faster in 2nd access,
20 ns faster in 3rd access, and
30 ns faster in 4th access.

As shown above, SDRAM has the same speed as EDO DRAM in 1st access, but the longer the burst length, the higher the data transfer speed SDRAM has.
Table 1-3 shows the access time of each SDRAM and EDO DRAM.

Table 1-3. Access Times of SDRAM and EDO DRAM

		1st (ns)	2nd (ns)	3rd (ns)	4th (ns)
SDRAM	143 MHz	42	49	56	63
	125 MHz	40	48	56	64
	100 MHz	50	60	70	80
	60 MHz	60	75	90	105
EDO DRAM	t_{RAC} = 60 ns	60	85	110	135
	t_{RAC} = 50 ns	50	70	90	110
	t_{RAC} = 40 ns	40	56	73	90

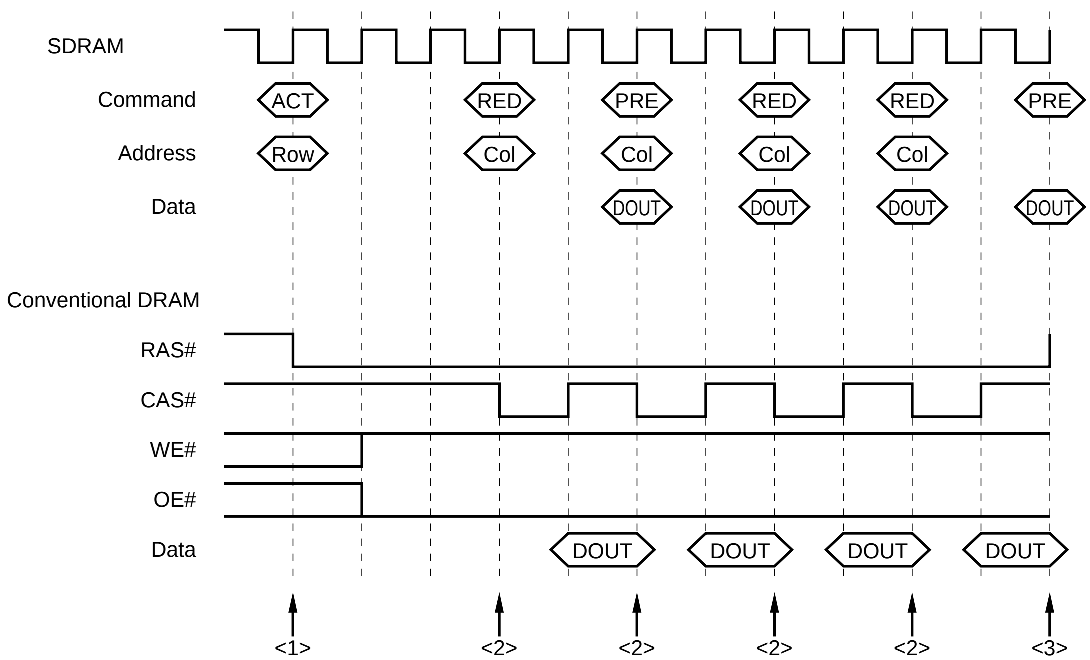
Table 1-3 shows only up to burst length = 4, but the longer the burst length, the larger the difference of performance between SDRAM and EDO DRAM.

(3) Using the conventional DRAM and SDRAM in the same cycle

The following explains an example of using the conventional DRAM and SDRAM in the identical cycle, taking the case of read cycle.

The read cycle of SDRAM in Figure 1-6 shows the case that CAS# latency = 2 (CL = 2), burst length = 1 (BL = 1).

Figure 1-6. Identical Read Cycle of Conventional DRAM and SDRAM



If a controller that generates the following signals is designed, the conventional DRAM and SDRAM can simultaneously be used.

Table 1-4. Correspondence of Control Signals of Conventional DRAM and SDRAM

Figure 1-6	Conventional DRAM	SDRAM
<1>	RAS# signal is active	Input active command
<2>	CAS# signal is active	Input read command
<3>	RAS# signal is inactive	Input precharge command

1.3 Control Method of SDRAM

The following explains control methods commonly used, taking 16-M SDRAM as an example. For the detailed control method of SDRAM, refer to the user's manual of SDRAM.

1.3.1 Pin functions

Table 1-5. List of Pin Functions

Pin Name	Input/Output	Pin Function
CLK (Input)	Input	CLK is the master clock input pin. Control signals of SDRAM are referenced to the CLK rising edge.
CKE (Input)	Input	CKE determines validity of the next CLK. If CKE is high, the next CLK rising edge is valid; otherwise it is invalid.
CS# (Input)	Input	CS# low starts the command input cycle.
RAS# (Input), CAS# (Input), WE# (Input)	Input	RAS#, CAS#, and WE# have the same symbols on conventional DRAM but different functions. For details, refer to **1.3.2 Commands**.
A0 to A11 (Input)	Input	A0 to A11 are address input pins. A10 defines the precharge mode. A11 is used for the bank select signal.
DQM (Input), UDQM (Input), LDQM (Input)	Input	DQM controls I/O buffers. DQM high and DQM low turn the output buffers off and on, respectively. ×16-bit products are capable of byte control (8-bit control). UDQM and LDQM control upper byte and lower byte input buffers, respectively
DQ0 to DQn	Input/Output	DQ0 to DQn are data I/O pins. DQn are DQ3, DQ7, and DQ15 in ×4-bit products, ×8-bit products, and ×16-bit products, respectively.
V_{CC}, V_{SS} (Power supply)	Input	V_{CC} and V_{SS} are power supply pins for internal circuits.

1.3.2 Commands

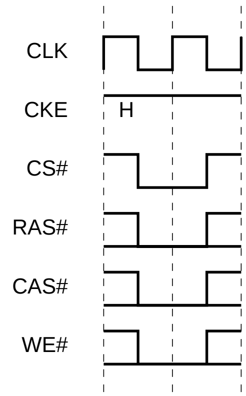
SDRAM, unlike the conventional DRAM, performs controls with commands. A command refers to a combination of logic levels of control signals. The following explains each command.

(1) Mode register set command

Mode register set command sets the mode register before entering the operation mode to set operation method in advance.

This command is executed with a combination of the logic level of each address signal pin. After power on, the mode register set command must be executed to set the mode register. The mode register retains the data until the setting is made again or the device loses power.

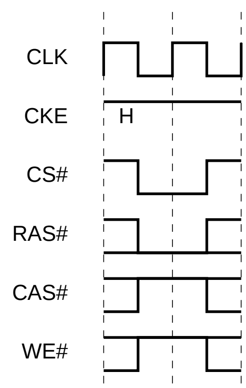
Figure 1-7. Mode Register Set Command



(2) Active command

Active command selects a bank with the address signal that controls the selection of bank and latches row address signal of the bank selected.

Figure 1-8. Row Address Strobe and Bank Active Command

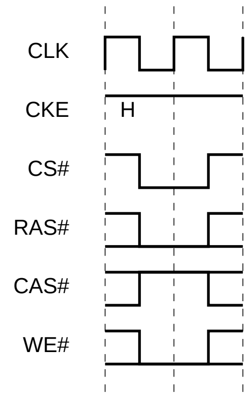


(3) Precharge command

Precharge command begins precharge operation of the bank selected.

When A10 is high, all the banks are precharged, regardless of the status of the address signal that controls the selection of bank. When A10 is low, only the bank selected by A11 signal is precharged.

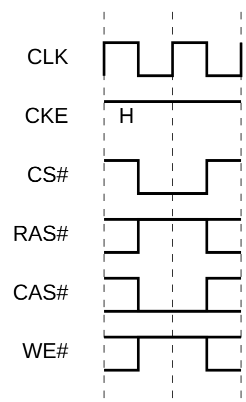
Figure 1-9. Precharge Command



(4) Read command

Read command begins read operation and latches the column address signal.

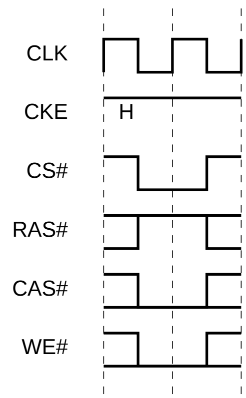
Figure 1-10. Column Address Signal and Read Command



(5) Write command

Write command begins write operation and latches the column address signal.

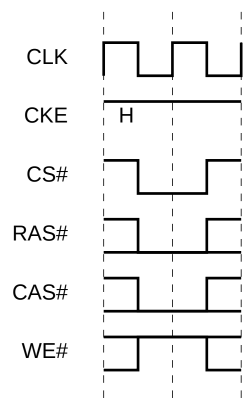
Figure 1-11. Column Address Signal and Write Command



(6) CBR (auto) refresh command

CBR (auto) refresh command executes CBR refresh operation. The refresh address signal is automatically generated internally. During certain period of time (t_{RC}) following this command, any other commands cannot be accepted.

Figure 1-12. CBR (Auto) Refresh Command

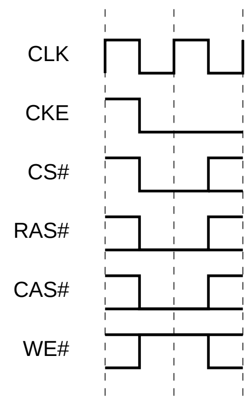


(7) Self refresh entry command

Self refresh entry command executes self refresh operation. Self refresh operation continues while CKE signal remains low. When CKE signal goes to high, the self refresh mode is terminated.

During self refresh mode, refresh interval and refresh operation are automatically performed internally, so there is no need for external control. During certain period of time (t_{RC}) following this command, the next command cannot be accepted.

Figure 1-13. Self Refresh Entry Command

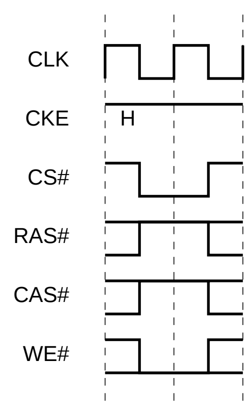


(8) Burst stop command

Burst stop command terminates the burst operation whose data is being transferred.

Although this command is a specification of the industrial standard, SDRAM of some companies do not support this command. Care should be taken when using this command.

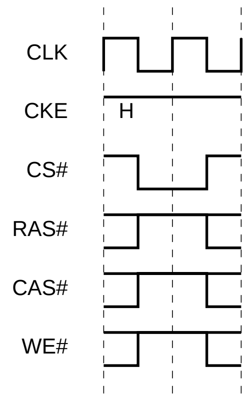
Figure 1-14. Burst Stop Command



(9) NOP command

NOP command does not perform any operations. No operations are started or terminated by inputting this command.

Figure 1-15. NOP Command



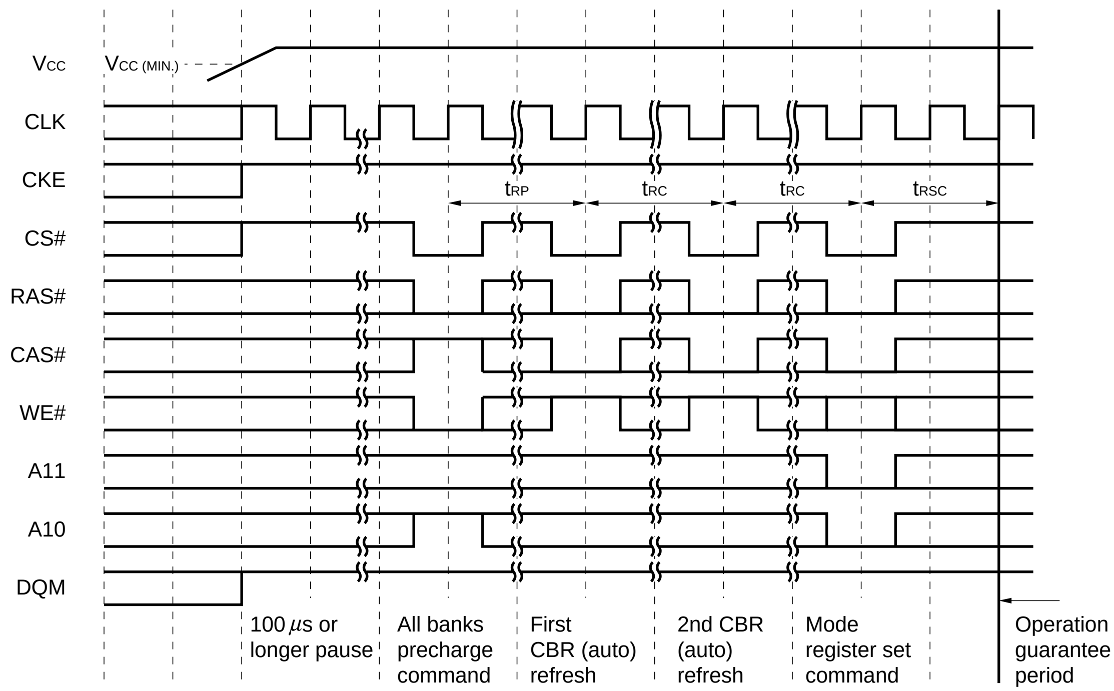
1.3.3 Initialization

SDRAM must be initialized in order to perform proper operation since the state of the circuit inside the SDRAM is unstable immediately after power on. SDRAM may not operate properly even during operation guarantee period unless initialization is performed properly.

To initialize internal circuit, precharge must be executed for both the banks using the all banks precharge command, etc. after a 100 μs or longer pause (high level of CKE# and CS# signals) when $V_{CC} \geq V_{CC\ (MIN.)}$ (after power supply voltage is stabilized). After precharge is completed, two or more CBR (auto) refresh must be performed.

After two or more CBR (auto) refresh are completed, the mode register can be set. The mode register set command can be input between the completion of precharge and the execution of CBR (auto) refresh.

Figure 1-16. Initialization of SDRAM



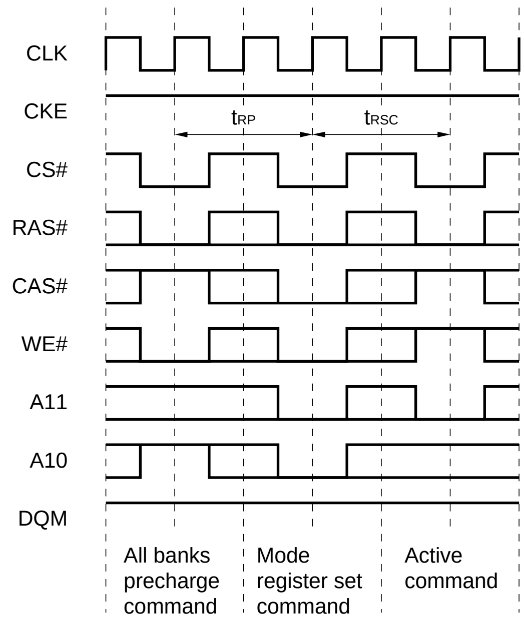
1.3.4 Setting of mode register

Mode register must be set to specify the operation method of SDRAM. The setting must be performed after power on before the actual operation starts.

Before setting the mode register, all banks (both banks) precharge command must be completed. And then, the mode register is set using the mode register set command.

To change the setting, the setting must be performed again. The resetting of the mode register can be performed even while SDRAM is operating.

Figure 1-17. Writing Mode Register



The following explains the contents of the mode register taking the case of 16-M SDRAM.

The mode register set command sets the operation of SDRAM according to the logic level of address (A0 to A11) signals.

The mode register has four fields:

(1) A0 to A2 : Setting of burst length
(2) A3 : Setting of wrap type
(3) A4 to A6 : Setting of CAS# latency
(4) A7 to A11: Options

(1) Setting of burst length

Setting of the number of words in which SDRAM continuously inputs/outputs data. The burst length is selectable as 1, 2, 4, 8, or full page.

Table 1-6. Setting of Burst Length

A2	A1	A0	When A3 = 0	When A3 = 1
0	0	0	1	1
0	0	1	2	2
0	1	0	4	4
0	1	1	8	8
1	0	0	Reserved	Reserved
1	0	1	Reserved	Reserved
1	1	0	Reserved	Reserved
1	1	1	Full page	Reserved

Generally, burst length = 4 is commonly used.
“Reserved” is set aside for the future extension of specifications. All the devices are prohibited to use the reserved.

(2) Setting of wrap type

The setting of the increment order of address signals in burst cycle.
When A3 = 0, the order is sequential, and when A3 = 1, the order is interleaving. The method chosen will depend on the type of DRAM controller used in each system. Generally, interleaving is commonly used.

(3) Setting of CAS# latency

The setting of the latency value after the read command is input until data is output.
The value of CAS# latency depends on the clock operating frequency and the speed grade of SDRAM.

Table 1-7. Setting of CAS# Latency

A6	A5	A4	CL
0	0	0	Reserved
0	0	1	Reserved
0	1	0	2
0	1	1	3
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

(4) Options

Secured for future extension of specifications. All of these addresses must be fixed to low.

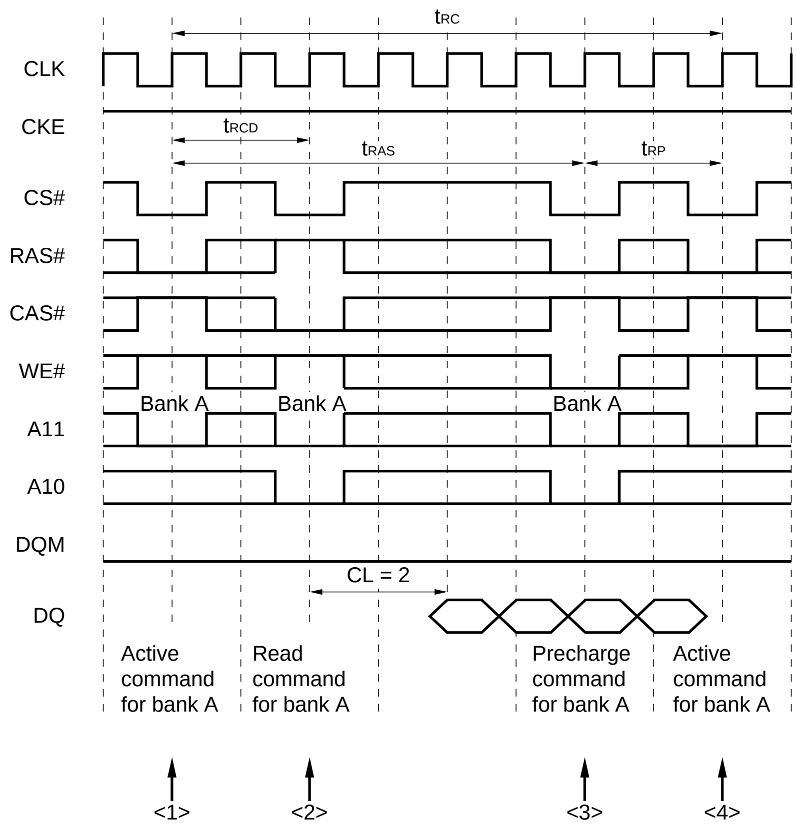
1.4 Operation Timing of SDRAM

The following explains the series of operation of SDRAM using each timing.

1.4.1 Read command and precharge

Figures 1-18 and 1-19 show the case that the read cycle is controlled with precharge when CAS# latency = 2 (CL = 2), burst length = 4 (BL = 4).

Figure 1-18. Read Command and Precharge (1)



At point <1>, an active command is input. When A11 is low, the active command controls bank A, and when A11 is high, it controls bank B (In Figure 1-18, it sets bank A).

At point <2>, (t_{RCD} after point <1>), a read command is input. When A10 is low, the read command controls precharge, and when A10 is high, it controls with auto precharge (In Figure 1-18, it sets precharge).

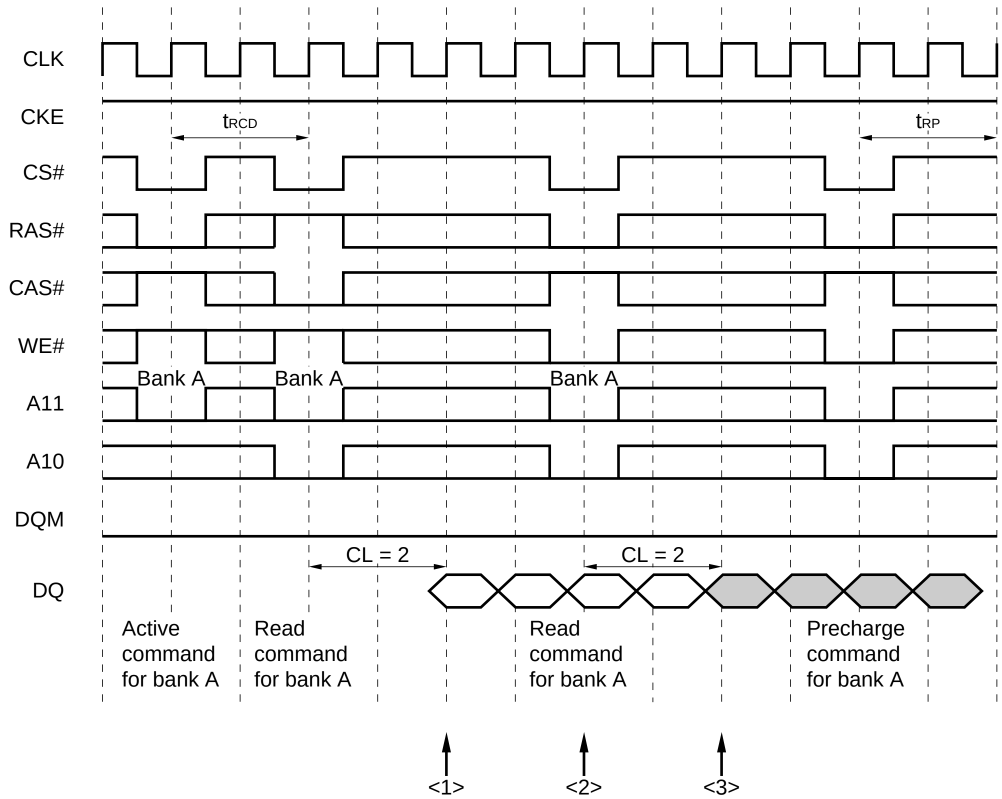
At point <3> (t_{RAS} after point <1>, or the point where the second data from the last is output. In Figure 1-18, whichever later of the points where the third data is output because BL = 4), a precharge command is input.

At point <4> (whichever later of t_{RC} after point <1> or t_{RP} after point <3>), an active command which indicates the next operation can be input. In Figure 1-18, the active command for bank A is input, but the active command for bank B can also be input.

Note that a precharge command is required at least t_{RP} before an active command is input.

To continuously output data of the same row address in the same bank, the data is output at point <3> if a read command is input at point <2>, so that the data can continuously be output.

Figure 1-19. Read Command and Precharge (2)



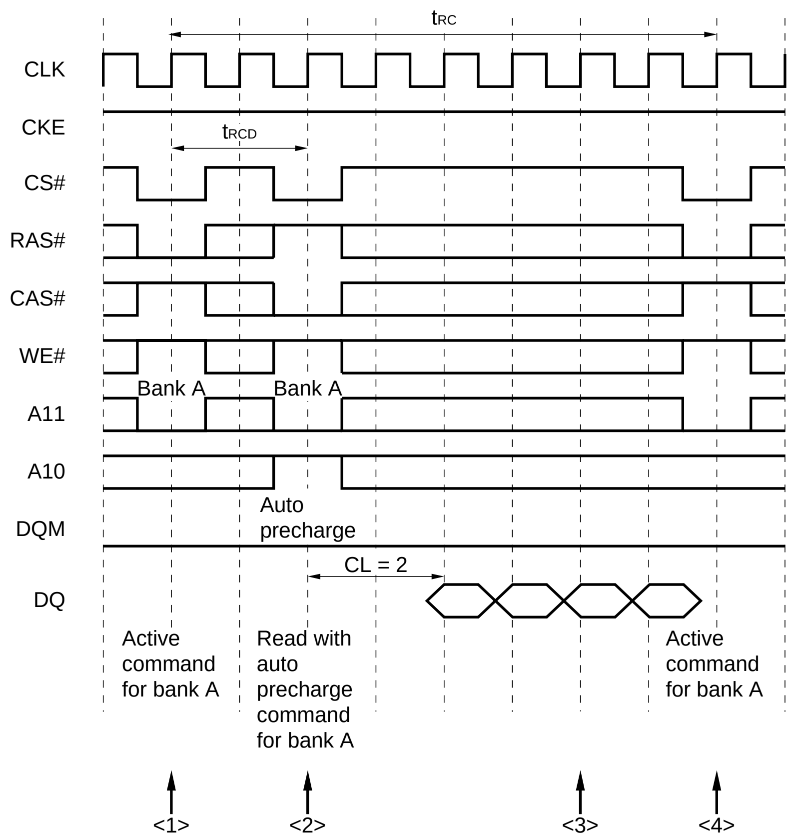
To output the data in the different bank (bank B in this case) from point <3> using the same control method as in Figure 1-19, input an active command for bank B at point <1>, and input a read command for bank B at point <2>.

When controlling different row address signals (different row address signals for bank A in this case), the data cannot continuously be output. The reason is if an active command for bank A with different row address signal at point <1> is input, a read command for bank A cannot be input at point <2>. As described earlier, a precharge command is required t_{RP} before an active command is input at point <1>.

1.4.2 Read command and auto precharge

Figure 1-20 shows the case that the read cycle is controlled with auto precharge when CAS# latency = 2 (CL = 2), burst length = 4 (BL = 4).

Figure 1-20. Read Command and Auto Precharge



At point <1>, an active command is input (In Figure 1-20, it sets bank A).

At point <2>, a read command is input (auto precharge in Figure 1-20).

At point <3>, a precharge command is automatically input inside the memory when the second data from the end of data is output (In Figure 1-20, the third data is output for BL = 4).

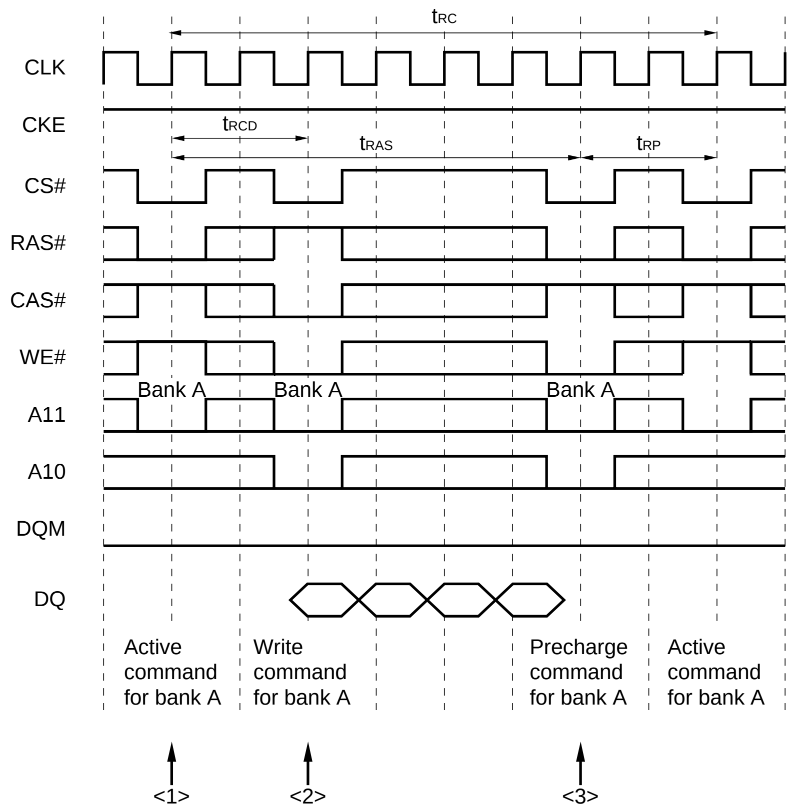
At point <4> (t_{RC} after point <1>), an active command which indicates the next operation is input.

For auto precharge, an active command is required at point <4> because a precharge command is automatically set at point <3>. When controlling the same row address signals or the same bank at points <1> and <4>, active commands are again required to set the same row address signal or the same bank as point <1> at point <4>.

1.4.3 Write command and precharge

Figures 1-21 and 1-22 show the case that the write cycle is controlled with precharge when burst length = 4 (BL = 4). In write cycle, all CAS# latency = 0 (CL = 0).

Figure 1-21. Write Command and Precharge (1)



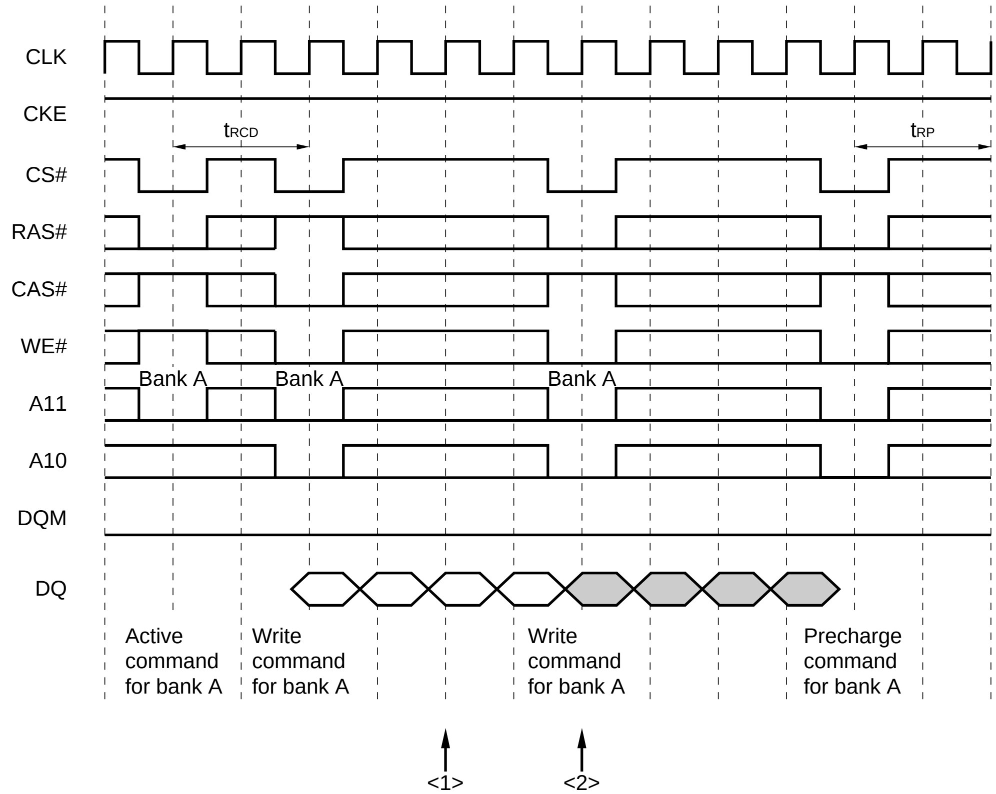
The basic controls in write cycle are the same as in read cycle. The differences are at points <2> and <3>.

In write cycle, data can be input immediately after a write command is input at point <2> (t_{RCD} after point <1>) because all CL = 0.

At point <3> (whichever later of t_{RAS} after point <1> or the point after the end of data input), a precharge command is input.

To continuously control data with the same row address in the same bank, the next data can be input from point <3> if a read command is input at point <3>, so that the data can continuously be input.

Figure 1-22. Write Command and Precharge (2)

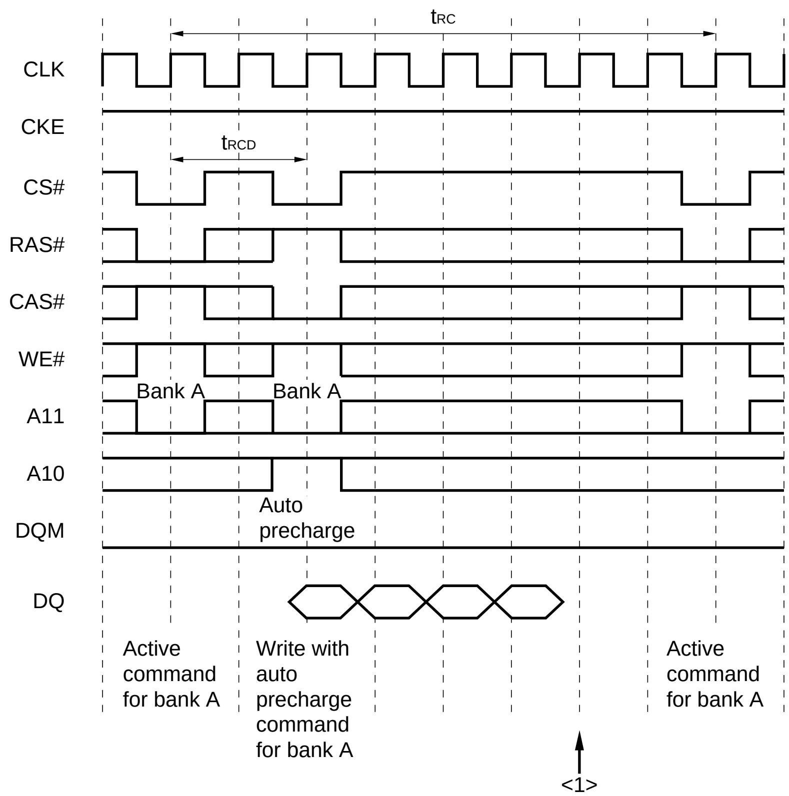


To input the data in the different bank from point <2> using the same control method as in Figure 1-22, input an active command for bank B at point <1>.

1.4.4 Write command and auto precharge

Figure 1-23 shows the case that the write cycle is controlled with auto precharge when burst length = 4 (BL = 4).

Figure 1-23. Write Command and Auto Precharge



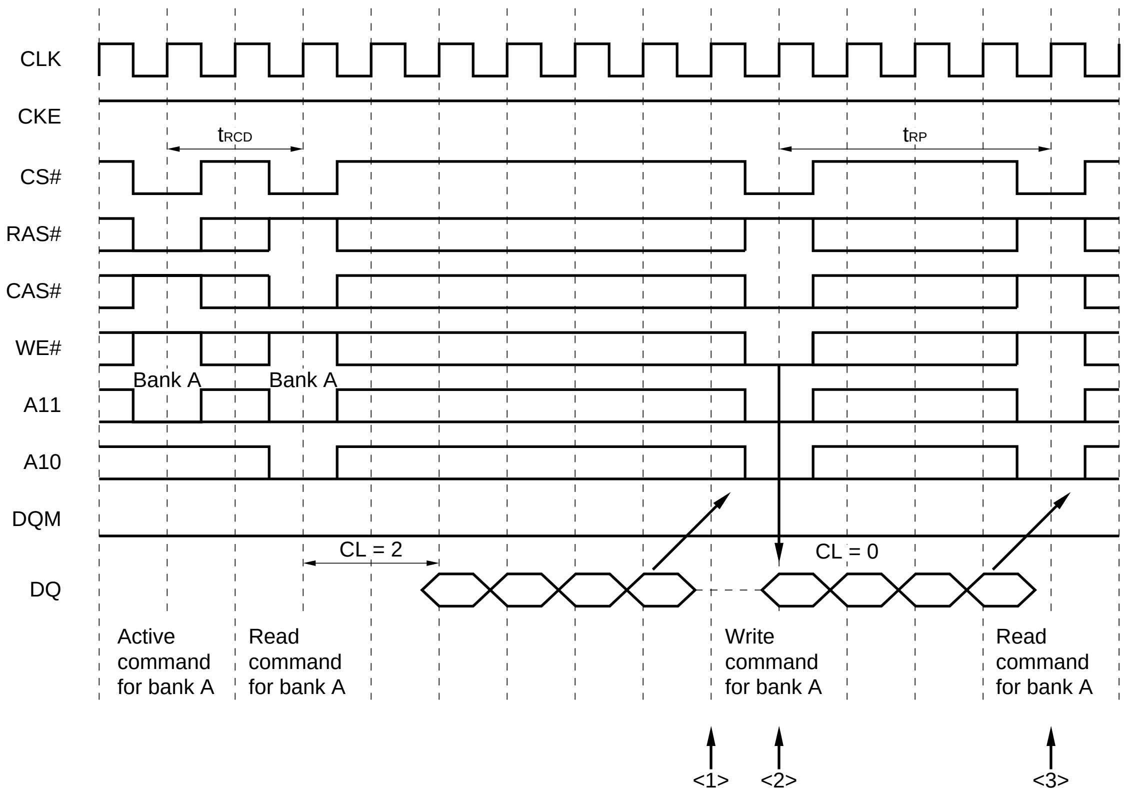
The basic controls in write cycle are the same as in read cycle.

A precharge command is automatically input inside the memory at the point after the last data of write cycle is input (point <1>).

1.4.5 Read command and write command

Figures 1-24 and 1-25 show the cases the read cycle and write cycle of the same bank (only bank A in Figure 1-24) are continuously executed when CAS# latency = 2 (CL = 2), burst length = 4 (BL = 4).

Figure 1-24. Read & Write Command (1)



At point <2>, a write command is input. The write command is input leaving one clock after the last data of the read cycle is output.

At point <1>, DQ must be set to high impedance state for one clock period in order to avoid bus fight.

At point <3>, a command that indicates the next operation (read command for bank B in Figure 1-24) is input.

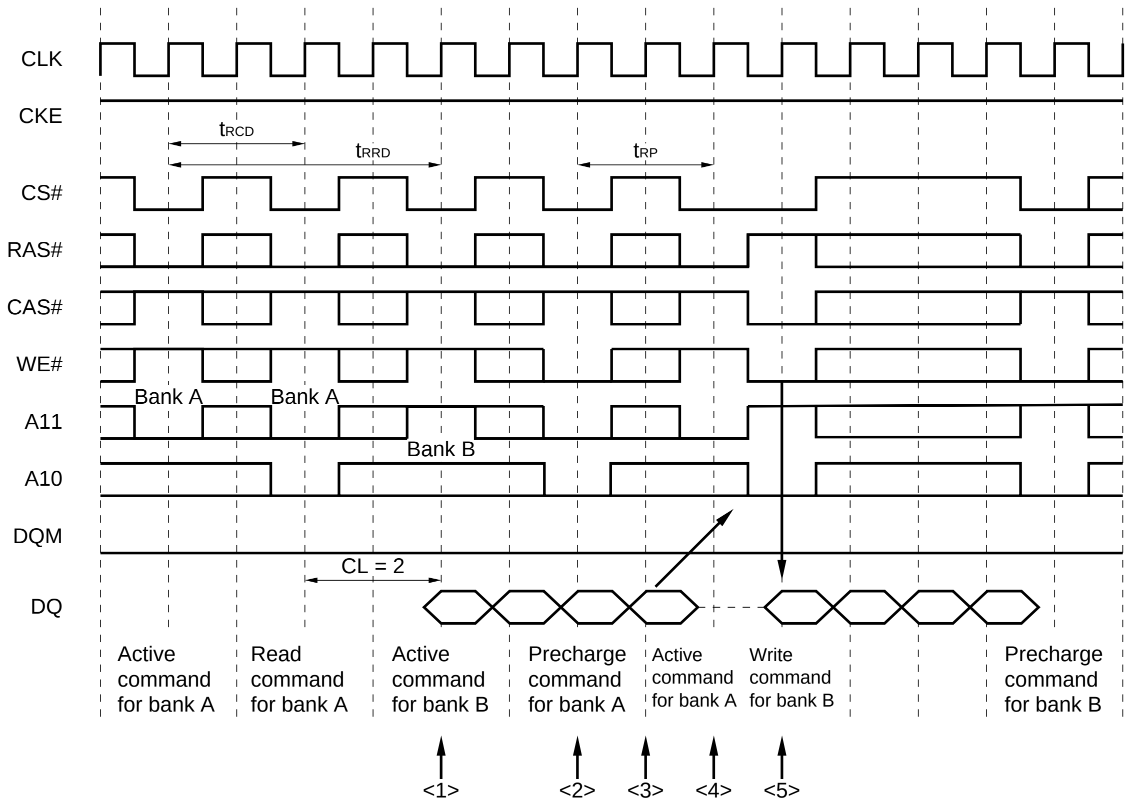
When changing the bank (from bank A to bank B in Figure 1-25), an active command is required between points <1> and <3> (input of an active command is not required when the bank is not changed).

An active command for bank B can be input t_{RRD} after the active command for bank A is input.

When inputting an active command for bank A at point <4>, a precharge command for bank A is required at point <2>.

The other command inputs are the same as for read and write command for the same bank.

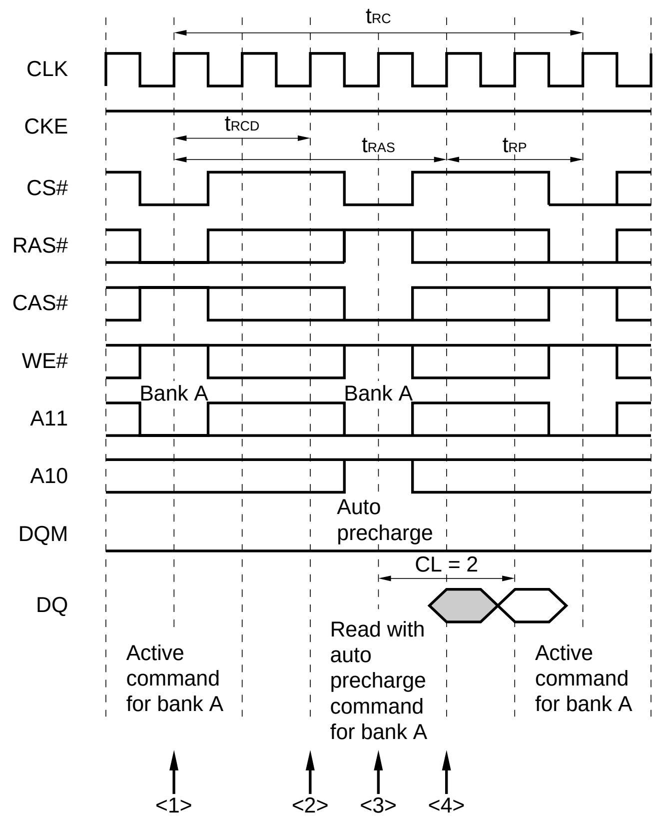
Figure 1-25. Read & Write Command (2)



1.4.6 Precautions for using access with burst length = 1

Figure 1-26 shows the case that the read cycle is controlled with auto precharge when CAS# latency = 2 (CL = 2), burst length = 1 (BL = 1).

Figure 1-26. Read Command and Auto Precharge



At point <1>, an active command is input.

If a read command is input at point <2>, data is output at point <4>. Since burst length is 1, a precharge command is automatically input at point <3>. In this case, the standard value of t_{RAS} is not observed, so that proper operation is not performed.

Therefore, in Figure 1-26, the control that delays the read command to point <3> and satisfies the standard value of t_{RAS} is required. In this case, a precharge command is automatically set at point <4>.

In the case of the write cycle, the operations are the same except CL = 0.

1.4.7 Merits and demerits of auto precharge

Table 1-8. Merits and Demerits of Each Type of Precharge

Precharge Method	Merits	Demerits
Auto precharge	A precharge command is automatically input if a precharge command is not input (Inputting precharge command is not required).	A precharge command is automatically input when t_{RAS} cannot be observed (Figure 1-26, etc.) or when a precharge command is not required (when accessing the same row address). A separate control is needed.
Precharge	A precharge command can be input at any timing t_{RAS} after the active command.	It must be judged whether a precharge command is required or not.

(1) Auto precharge

In auto precharge, a precharge command is automatically input inside the SDRAM and not required to be input from the external of the memory. When t_{RAS} cannot be observed as shown in Figure 1-26, however, the control that delays the auto precharge command is separately required. In addition, when continuously accessing the same row address signal, it is essentially not required to re-input an active command. In auto precharge, it is required to re-input an active command of the same row address signal because a precharge command is automatically input.

Considering the circumstances shown above, the use of auto precharge is convenient for the control in which the burst length is fixed and an active command is input whenever the burst ends.

(2) Precharge command

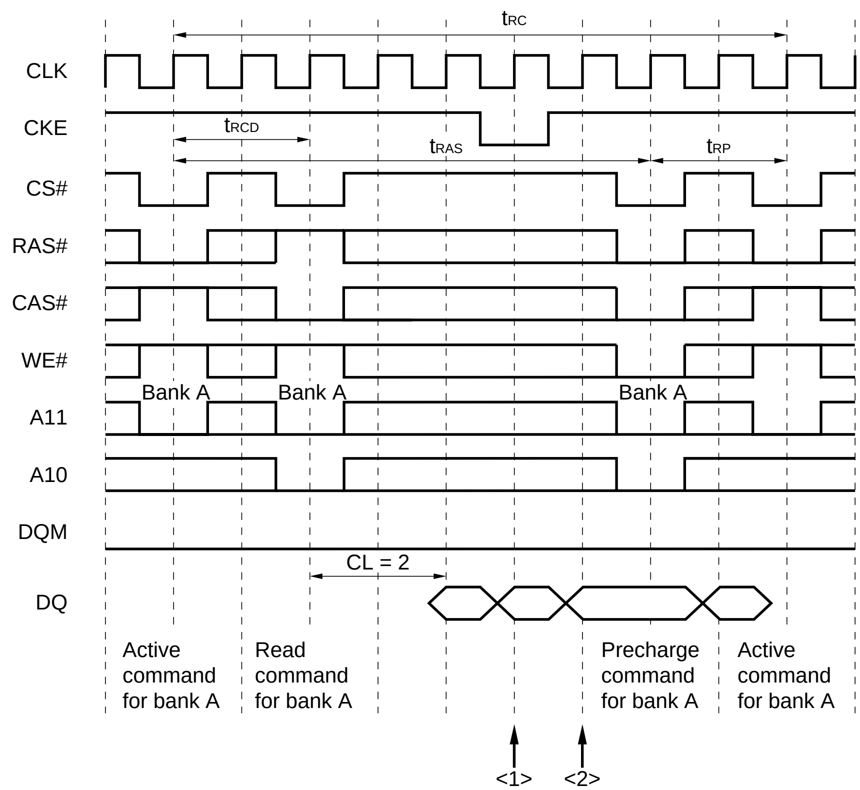
In precharge, a precharge command can be input at any timing t_{RAS} after the active command. For example, the use of precharge is convenient for the control in which the burst length is not fixed but fluctuated depending on the cycle (i.e., burst length = 4 in read cycle and burst length = 1 in write cycle) and for the control which has less chances of changing row address signals.

1.4.8 Temporary stop of clock during data transfer

Figure 1-27 shows the case that the read cycle is controlled with precharge when CAS# latency = 2 (CL = 2), burst length = 4 (BL = 4).

Figure 1-28 shows the case that the write cycle is controlled with the same condition as above.

Figure 1-27. Temporary Stop of Clock in Read Cycle



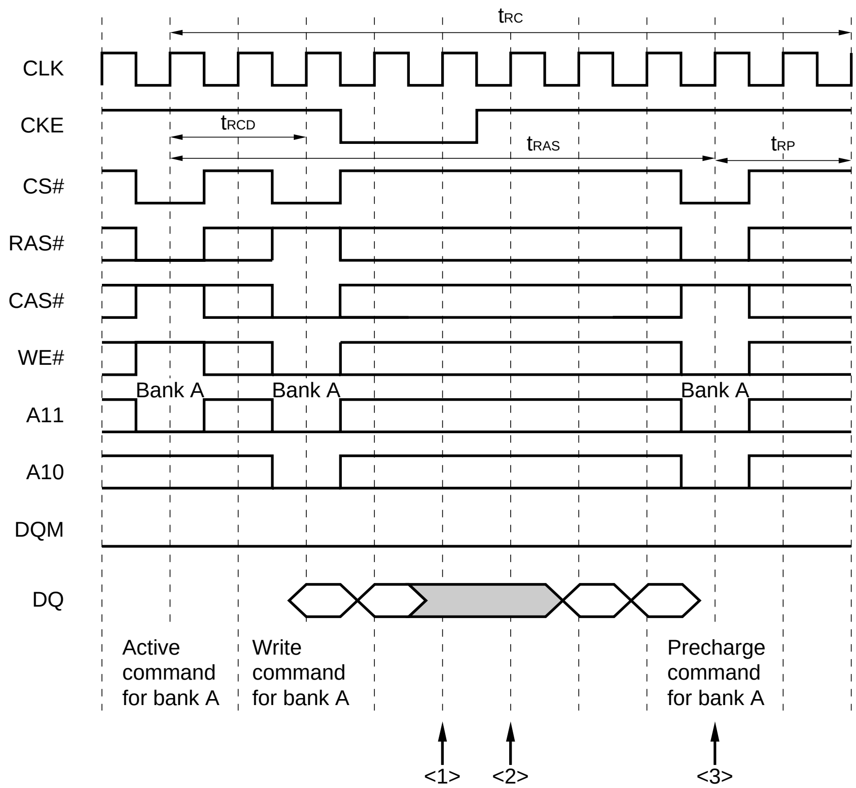
The internal clock of memory can be stopped by setting CKE signal to low at point <1>. The internal clock of memory stops at point <2>. The latency from CKE signal to the stop of the internal clock of memory is always 1 regardless of the number of CL.

In Figure 1-27, CLK signal is always input. The internal CLK signal of memory, however, stops for one clock at point <2>.

t_{RC} and t_{RAS}, therefore, delays for one clock when the external CLK signal is taken as a reference.

In Figure 1-28, the clock is temporarily stopped for two clocks in the write cycle.

Figure 1-28. Temporary Stop of Clock in Write Cycle

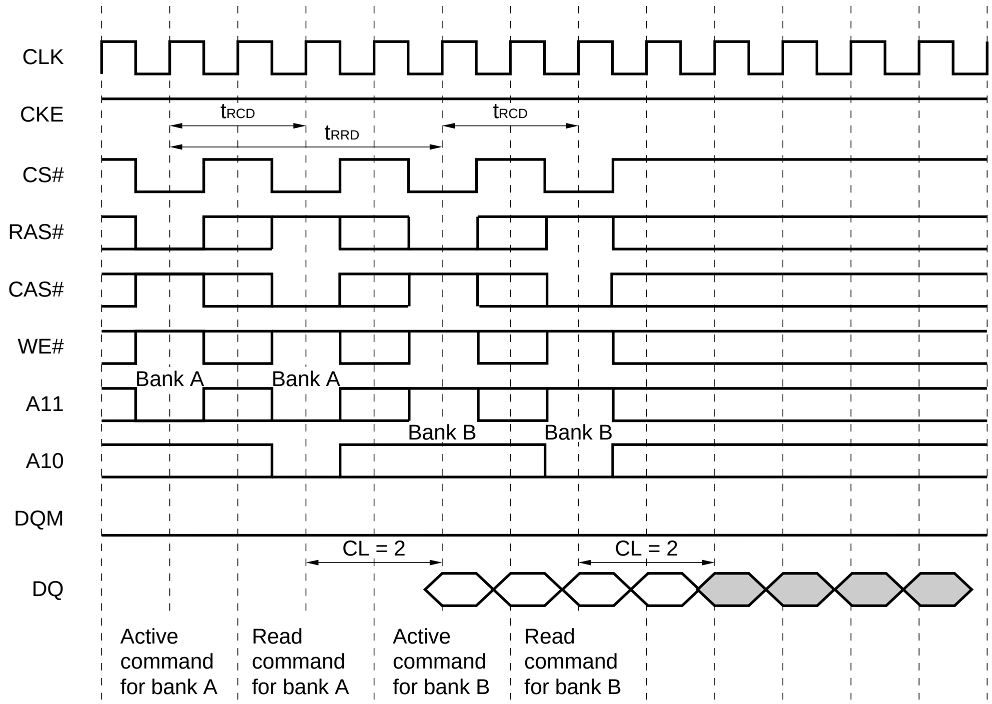


The clock is being stopped at points <1> and <2>. The precharge command, therefore, delays for two clocks from the external CLK signal.

1.4.9 Both-bank ping-pong control

Figure 1-29 shows the case that the read cycle is continuously and alternately controlled with different banks when CAS# latency = 2 (CL = 2), burst length is 4 (BL = 4).

Figure 1-29. Both-bank Ping-pong Read



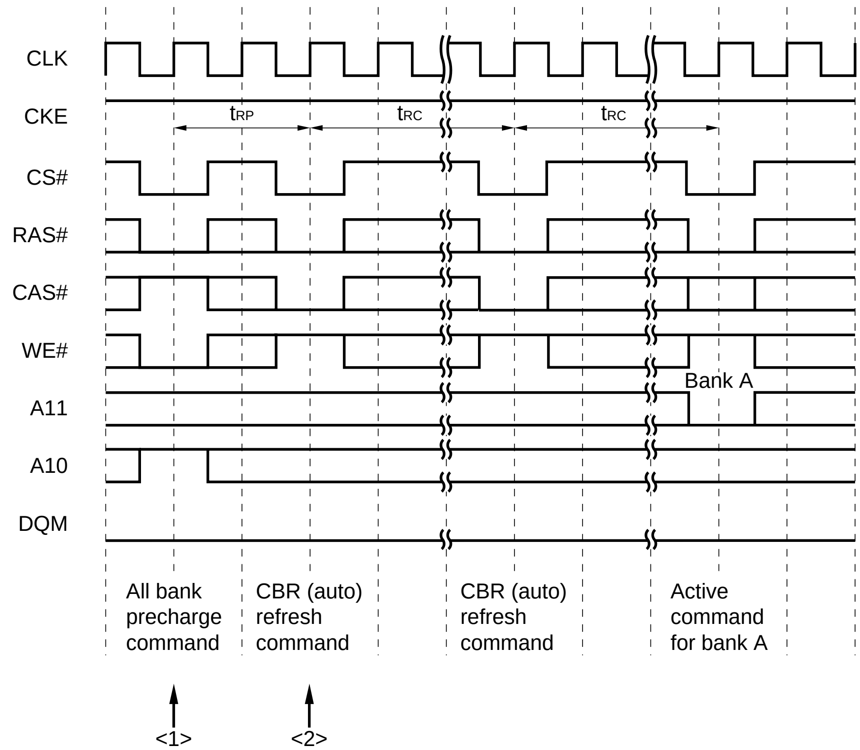
Active commands for different banks can alternately (ping-pong) be input. Ensure that the standard value of $t_{RAS\ (MAX.)}$ is satisfied.

In the case of the write cycle, the operations are the same except CL = 0.

1.4.10 CBR (auto) refresh

Figure 1-30 shows the case that two CBR (auto) refresh commands are input consecutively. Two CBR commands do not necessarily have to be input consecutively.

Figure 1-30. CBR (Auto) Refresh



When inputting a CBR (auto) refresh command at point <2>, an all bank precharge command must be input t_{RP} before inputting CBR (auto) refresh command (point <1>).

An active command to indicate the next operation can be input t_{RC} after CBR (auto) refresh command is input.

CHAPTER 2 PCI BUS

This chapter provides an outline of PCI bus.

The specifications of PCI bus is subject to change without notice. Consult PCI Special Interest Group to acquire the latest specifications of the PCI bus before starting designing work.

2.1 Outline of PCI Bus

2.1.1 PCI bus

Currently, the PCI bus is widely adopted as a high-speed I/O bus interface. The main stream of data transfer time of the PCI bus is 132 Mbytes/s, and it operates in synchronization with a bus clock of 33 MHz.

The PCI bus requires a device for the bridge (PCI bridge) between the local bus and the PCI bus. The use of this bridge makes the PCI bus a versatile high-speed bus that does not depend on the performance or type of CPU.

PCI bus rapidly penetrated the market along with the popularization of Pentium™ processor. The PCI bus is currently the mainstream of high-speed busses.

2.1.2 Explanation of pins

The following explains typical pins of the PCI bus. The descriptions here do not cover all the pins of the PCI bus.

The bus master arbitrates with other devices, and regularly performs controls. The target is a device or unit to perform the operation that the bus master directed .

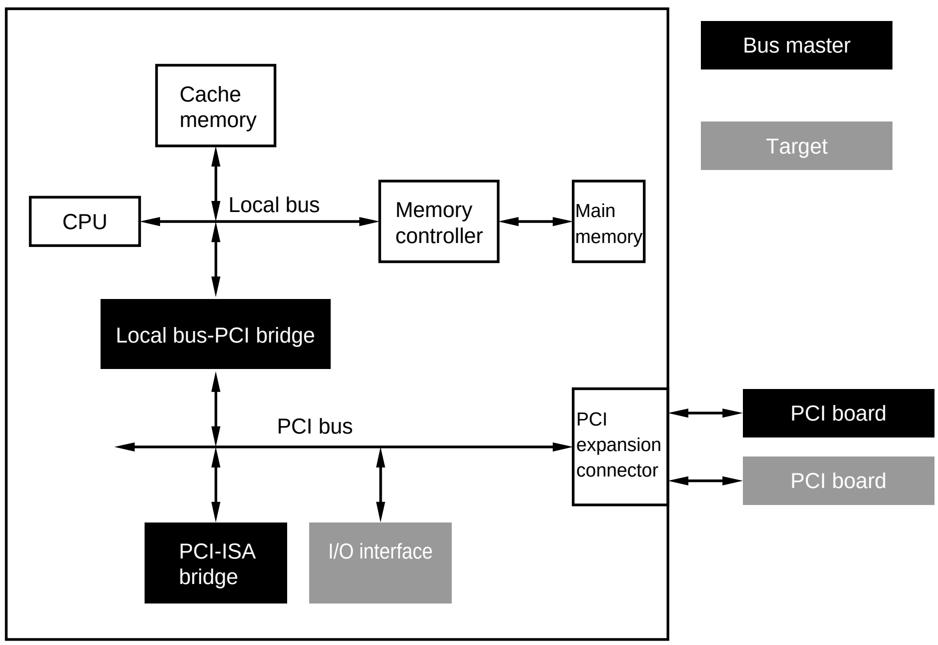
Table 2-1. Signal Table

Signal	Description
CLK signal (Clock)	Signal to be input to the target. This signal is referenced on the bus. 33 MHz and 66 MHz are defined in version 2.0 and 2.1 of PCI bus, respectively.
RST# (Reset)	Signal to be input to the target. This signal initializes all the target. RST# low from the bus master initializes all the target.
AD (Address signal and data)	Multiplexed signal of address signal and data. Address is a signal to be output to the target, and data is a signal input/output to/from the target. Address signal is processed first (address phase), and data is processed with time difference (data phase).
C/BE [3 : : 0] # (Bus command and byte enable)	Signal to be input to the target. This is a multiplexed signal for bus command and byte enable. It controls bus command and byte enable in address phase and data phase of AD, respectively.
FRAME# (Cycle frame)	Signal to be input to the target. This signal indicates the period in which an access is executed. FRAME# low indicates the bus is controlling data transfer.
IRDY# (Initiator ready)	Signal to be input to the target. This signal becomes low when the bus master is ready for read or write operation. It is used simultaneously with TRDY# and becomes wait state when either is high and becomes data transfer state when both are low.
TRDY# (Target ready)	Signal to be output from the target. This signal becomes low when the target is ready for read or write operation. It is used simultaneously with IRDY# and becomes wait state when either is high and becomes data transfer state when both are low.
STOP# (Stop)	Signal to be output from the target. This signal is output from the target to request the bus master to stop the current transfer.
DEVSEL# (Device Select)	Signal to be output from the target. This is a response signal to the signal with which the target directs the bus master to start operation.
REQ# (Request, dedicated signal for bus master)	Signal with which a certain bus master requests the use of the bus from the bus master (arbiter) which arbitrates the bus when there are more than one bus masters.
GNT# (Grant, dedicated signal for bus master)	Signal with which the bus master (arbiter) that arbitrates the bus permits the use of the bus to the bus master that has requested the use of the bus.

2.1.3 Block diagram of PCI bus interface

There are two types of signals for the control signals of the PCI bus: output signal and input signal. Taking the case of output signal, whether the bus master or the target outputs the signal must be clarified. In the same way, whether the bus master or the target inputs signal must be clarified in the case of input signal.

Figure 2-1. Bus Master and Target



The bus means the PCI bus in this case. The bus master arbitrates with devices other than the one using the bus and regularly controls the bus. Any device or unit can be the bus master as long as it is capable of arbitrating and controlling the bus. In Figure 2-1, major bus master is a bridge, etc. PCI card can also be the bus master if it is capable of arbitrating with each bridge and regularly controlling the bus.

The target is a device or unit used by the bus master, and it starts operation when directed from the bus master.

2.1.4 Bus command

Bus command sets in advance the destination of data transfer in address phase. Bus command performs different controls depending on the combination of the four logic levels of C/BE#.

Table 2-2. Bus Command

C/BE3#	C/BE2#	C/BE1#	C/BE0#	Controls
0	0	0	0	Interrupt acknowledge
0	0	0	1	Special cycle
0	0	1	0	I/O read
0	0	1	1	I/O write
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	Memory read
0	1	1	1	Memory write
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Configuration read
1	0	1	1	Configuration write
1	1	0	0	Memory read multiple
1	1	0	1	Dual address cycle
1	1	1	0	Memory read line
1	1	1	1	Memory write and invalidate

Of the bus commands shown in Table 2-2, the circuit used in CHAPTER 3 (refer to **Figure A-1 Entire Circuit Diagram**) uses memory read and memory write commands. This section, therefore, explains only memory read and memory write commands.

(1) Memory read command

This is a command with which the bus master declares the reading data from a memory mapped in the memory address space.

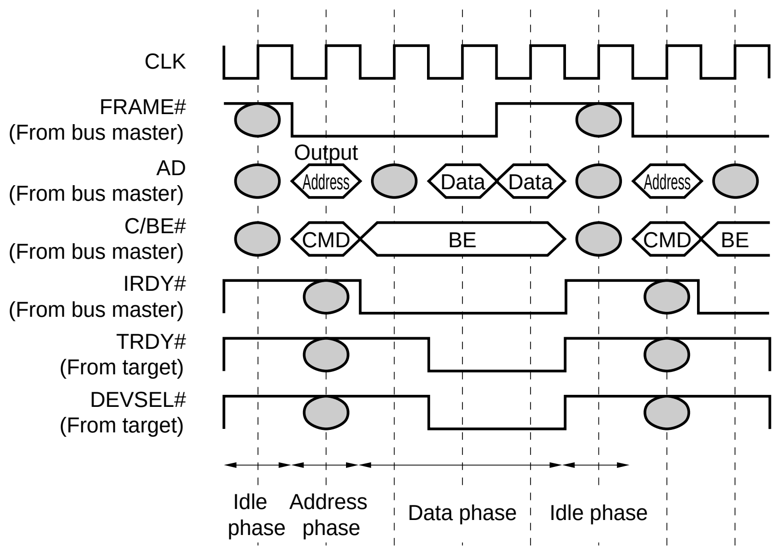
(2) Memory write command

This is a command with which the bus master declares the writing data in a memory mapped in the memory address space.

2.1.5 Bus drive and turn-around cycle

All the signals controlled with more than one bus masters requires turn-around cycles. The turn-around cycle is a cycle to set aside a certain interval to avoid the collision of signals when one bus master stops the drive of a device and another bus master starts the drive of that device. The shadowed portion in Figure 2-2 indicates turn-around cycles.

Figure 2-2. Turn-around Cycle of Each Phase



(1) IRDY#, TRDY#, DEVSEL#, etc. require turn-around cycles in address phase.

(2) FRAME# and C/BE# require turn-around cycles in idle phase.

Idle phase is a state where both FRAME# and IRDY# are high.

(3) AD changes the direction of input/output because the address and data are multiplexed. Turn-around cycles, therefore, are required at different phases.

- When inputting data, turn-around cycles are required in idle phase and address phase.
- When outputting data, turn-around cycles are required only in idle phase.

2.2 Data Transfer

2.2.1 Outline of data transfer

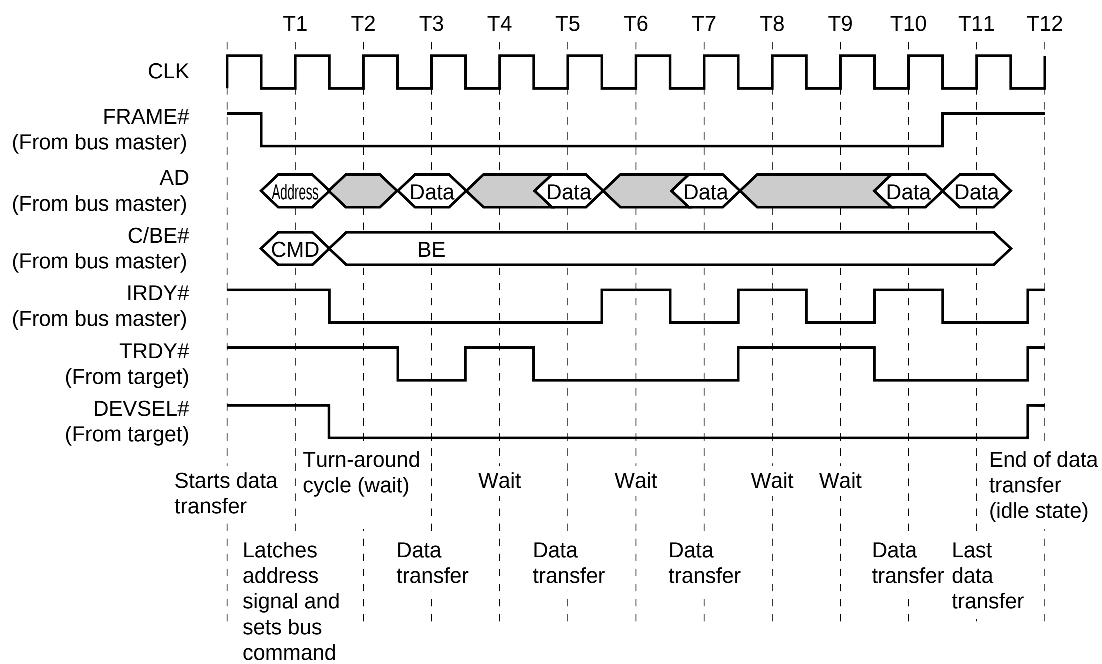
Data transfer is started when FRAME# is low.

In read cycle, address signal is output and data is input. Turn-around cycles, therefore, must be inserted between address signal and data. In write cycle, turn-around cycles are not required because both address signal and data are output.

When FRAME# is low and either or both of IRDY# or/and TRDY# is/are high, a wait is inserted. IRDY# high indicates that the bus master is requesting a wait from the target. TRDY# high indicates that the target is requesting a wait from the bus master.

When both IRDY# and TRDY# becomes low at the same time while FRAME# is low, data transfer starts. When both IRDY# and TRDY# become low at the same time while FRAME# is high, this indicates the last data transfer and idle state must be set afterward.

Figure 2-3. Basic Cycle



Controls of read cycle and write cycle are performed with command (CMD).

2.2.2 End of data transfer

The end of data transfer can be controlled either with bus master or target. In neither case, data transfer cannot one-sidedly be ended. The bus master has the final authority and orderly controls the end of data transfer.

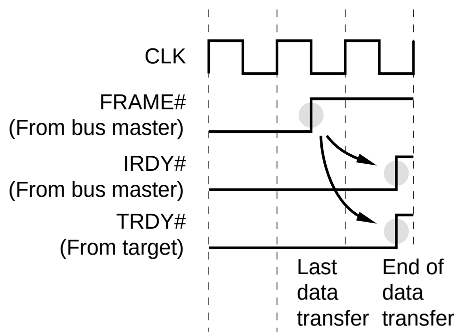
There are several methods to end data transfer. This section explains only basic processings.

(1) When controlled with the bus master

When the bus master sets FRAME# to high while both IRDY# and TRDY# are low, it indicates that the data phase of the next clock is the last data transfer. The bus master notifies the target that the data transfer is ending.

When both IRDY# and TRDY# are high, data transfer is completed.

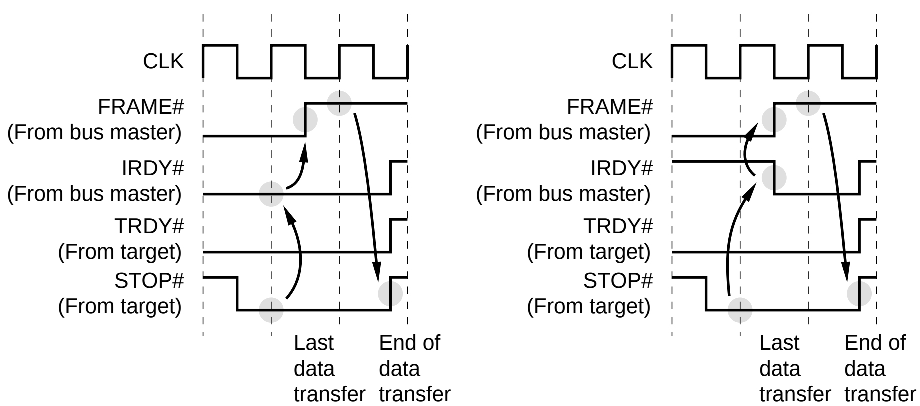
Figure 2-4. End of Transfer Controlled with Bus Master



(2) When controlled with the target

The target sets STOP# to low and requests the bus master to end data transfer.

Figure 2-5. End of Transfer Controlled with Target



After STOP# is set to low, the bus master immediately sets FRAME# to high to end data transfer. Once STOP# is set to low, do no set STOP# to high until FRAME# becomes high.

2.2.3 Arbitration

When there are more than one bus masters, they use REQ# and GNT# to orderly perform controls.

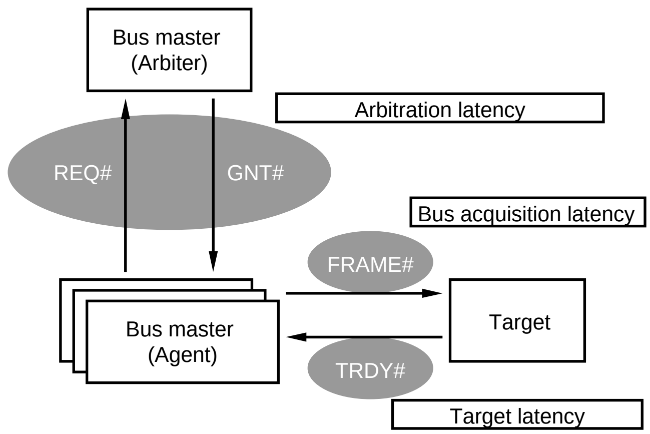
The bus masters request the use of bus by setting REQ# to low. Do not set REQ# to low unless the bus is actually used. REQ# and GNT# are dedicated signals for the bus masters.

The bus master (arbiter) which arbitrates the bus sets GNT# to low when it judges that one of the bus masters (agent, etc.) which are to control given processing can use the bus.

2.2.4 Latency

Latency is the time after one bus master requests the use of bus until the target inputs/outputs the data. Figure 2-6 includes three latencies.

Figure 2-6. Access Latency



(1) Arbitration latency

The period when more than one bus masters arbitrates the bus using REQ# and GNT#.

The time after one bus master sets REQ# to low until the bus master that arbitrates the bus sets GNT# to low. This is 2-clock period for the bus master with the highest priority.

(2) Bus acquisition latency

The period when the bus masters have to wait until the bus is released.

The time after one bus is permitted to use the bus until it sets FRAME# to low. This is a 1-clock period.

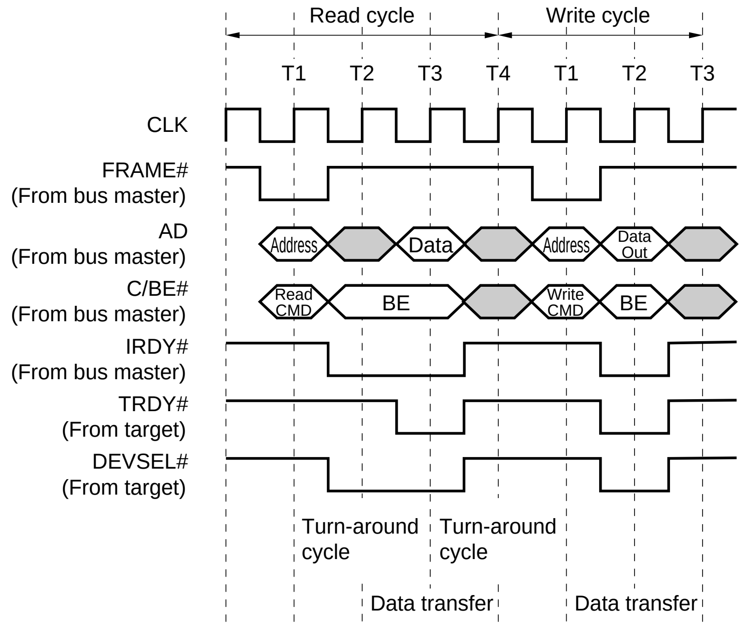
(3) Target latency

The period until the target sets TRDY# to low for the first data transfer. This is a 2-clock period in read cycle and a 1-clock period in write cycle.

2.2.5 Random access

In read cycle, turn-around cycles must be inserted between address signals and data inputs (in write cycle, turn-around cycles are not required). Compared to write cycle, therefore, read cycle has more clocks (read cycle consists of four clocks while write cycle consists of three clocks).

Figure 2-7. Random Access Cycle

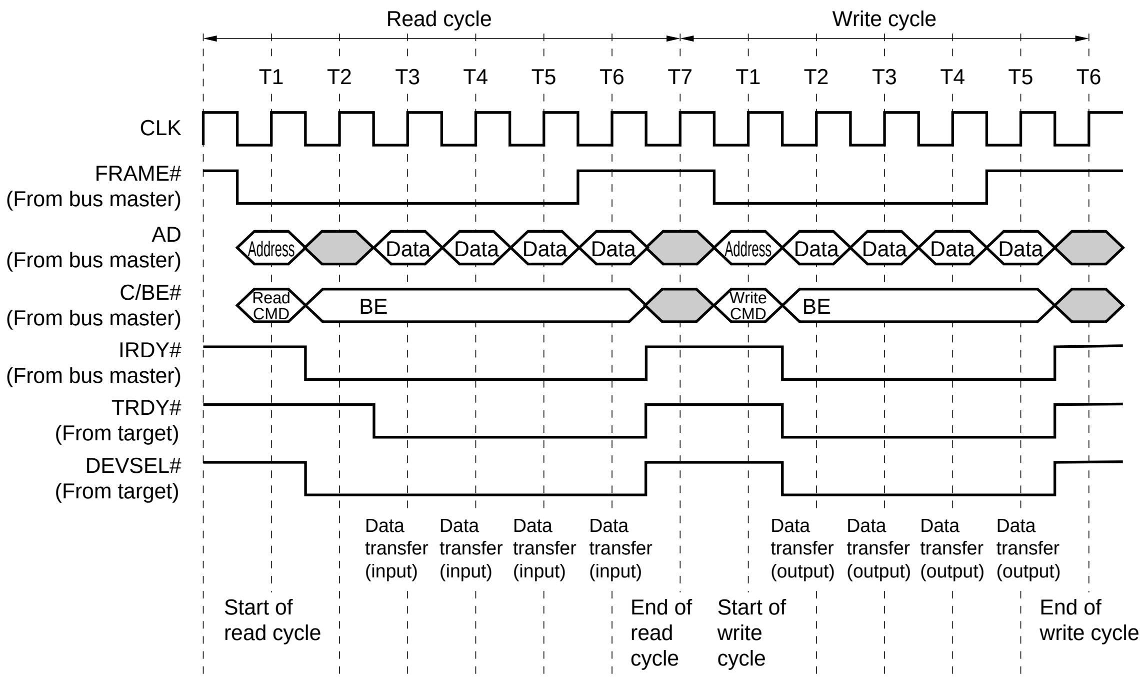


2.2.6 Burst access

In burst access, turn-around cycles must be inserted similarly to random access (in write cycle, turn-around cycles are not required). Compared to write cycle, therefore, read cycle has one more clock.

There is no limitation for the number of burst access (burst length = 4 in Figure 2-8). The length of the data, however, must be the same between the device to send the data and the device to receive the data. The length of the data is commonly burst length = 4.

Figure 2-8. Burst Read Cycle and Burst Write Cycle



CHAPTER 3 DESIGNING SDRAM CONTROLLER

This chapter describes the designing of the SDRAM controller to be connected to the PCI bus interface.

The memory mentioned is specified to SDRAM. Centering on SDRAM, the outline and points of designing a SDRAM controller that is required to connect the PCI slot and SDRAM. Descriptions such as read and write operations mean the read and write operations of SDRAM, and the operations from devices other than SDRAM such as PCI controllers are described as reverse operations.

As examples of circuit design of SDRAM controller, circuit diagrams are added in the appendix (refer to **Figure A-1 Entire Circuit Diagram**). The circuit is used for explanation hereafter.

The descriptions in this chapter show only the concept of operation and not actual operation. Use the descriptions only for reference purpose upon designing.

3.1 Outline of SDRAM Controller

CPU and SDRAM cannot be connected directly. This is because the CPU and SDRAM have different control signal functions and processing methods. The CPU is designed to be connected not only to SDRAM but to other memory and I/O equipment. Therefore, a memory controller between the CPU and SDRAM is required in order to connect them.

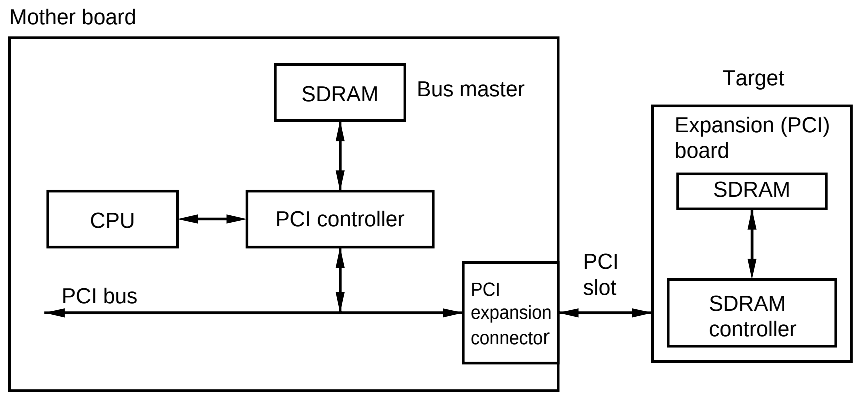
Most personal computers are provided with a CPU, memory, and a chip set. A chip set is configured with controllers of various devices, and a memory controller is also a part of the chip set.

Since SDRAM controllers that deal with controls of SDRAM only are explained in this case, the circuit examples are configured only with memory controllers.

3.2 Outline of Connection between SDRAM and PCI Bus

Figure 3-1 shows an example of outline of the PCI interface block diagram. In this figure, an expansion board is connected to the PCI slot on the mother board, and the SDRAM and SDRAM controller are configured in the expansion board. In this case, the expansion board and the PCI controller become the target and the bus master, respectively.

Figure 3-1. Block Diagram of PCI Interface



When transferring data from the PCI controller to SDRAM, the SDRAM controller generates the signal to control SDRAM using the signal output from the PCI controller and input the generated signal to SDRAM.

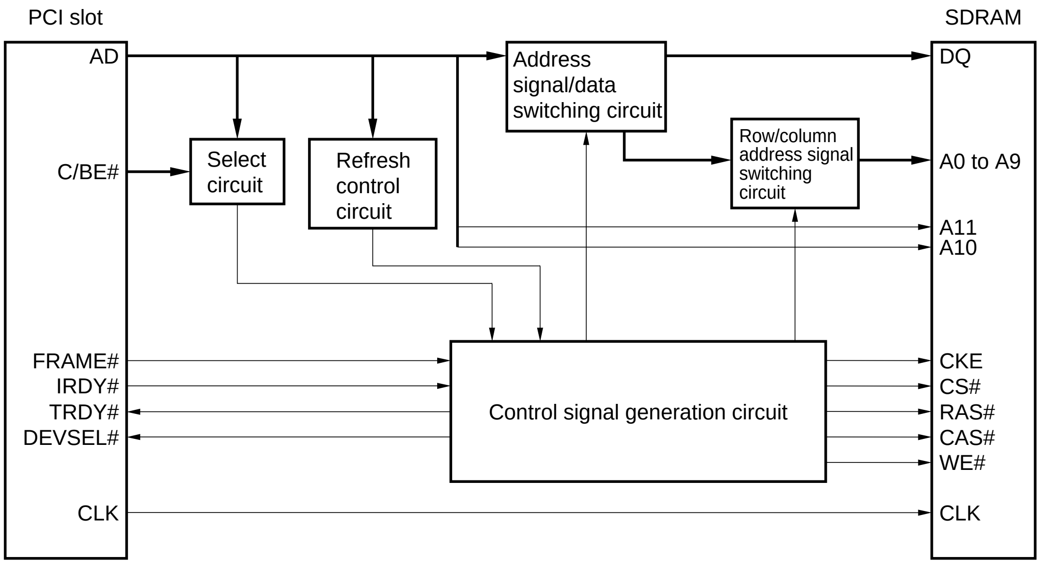
3.2.1 Block diagram of SDRAM controller

An SDRAM controller can be divided roughly into five blocks.

- Select circuit
- Refresh control circuit
- Address signal/data switching circuit
- Row/column address signal switching circuit
- Control signal generation circuit

This User's Manual uses two DRAM products with 16-Mbit, ×16-bit configuration. This is because the PCI bus has a 32-bit bus width, and the two SDRAM simultaneously perform identical operations.

Figure 3-2. Block Diagram of SDRAM Controller



(1) Select circuit

A circuit to Direct the control signal generation circuit to generate control signals required for read and write operations using the signals generated utilizing the change of address signals from the PCI controller. This circuit also defines which SDRAM is to operate when there are more than one SDRAM.

(2) Refresh control circuit

A circuit to generate the signals to control the refresh operation of SDRAM.

Refresh operation is controlled with software using memory space (the details are explained in **3.2.3 Memory space**).

Software selects memory refresh space set in advance in the memory space of the CPU after a certain period of time elapses (during refresh period). This circuit generates start signal of refresh operation with the change of address signal when the memory refresh space is selected and Direct control signal generation circuit to generate control signals required for refresh operation.

(3) Address signal/data switching circuit

This circuit performs the control of separating the multiplexed signal of address and data output from the PCI controller into address signal and data when inputting it to SDRAM. This is because SDRAM does not multiplex address signal and data while the PCI bus specification does.

(4) Row/column address signal switching circuit

This circuit performs the control of separating the address signal output from the PCI controller into row address and column address when inputting it to SDRAM. This is because SDRAM adopts address-multiplexed system for the address signal input method while the address signal output from the PCI controller does not support this system.

(5) Control signal generation circuit

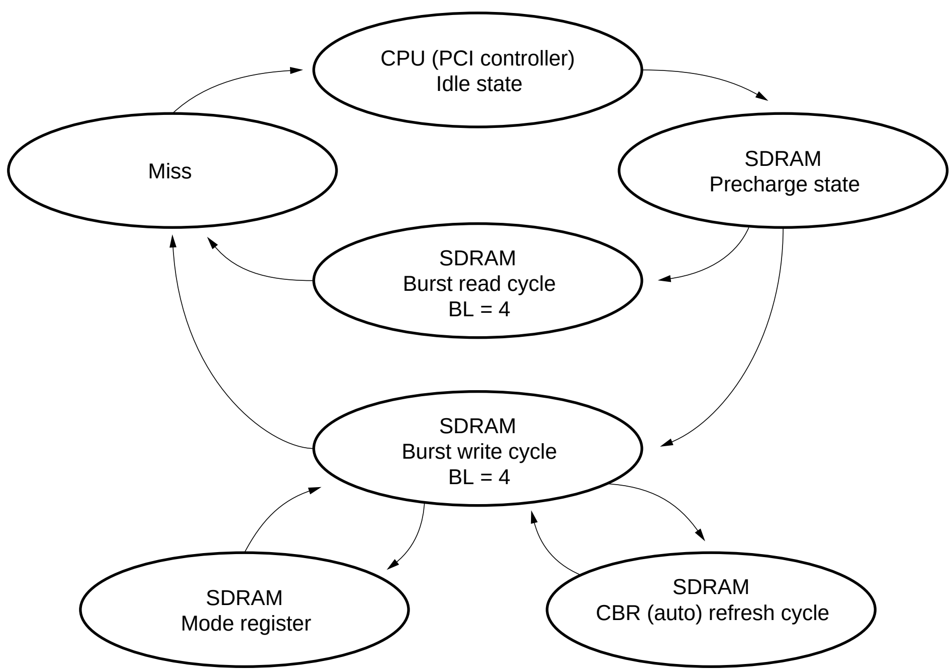
This is the most important circuit in Figure 3-2. It generates all the control signals of SDRAM. For example, when the control signal generation circuit receives directions for various controls from the select circuit, it immediately generates the control signal to control SDRAM.

3.2.2 State diagram of SDRAM controller

The following explains, using the state diagram, the method by which the SDRAM controller controls SDRAM.

In the circuit example, the control with burst length = 4, which is the most basic burst length. Burst length = 1 and burst length = 4 can coexist to perform controls. When the same row address signal is controlled (hit), it is not necessary to input the row address (miss) again. In this circuit example, however, miss is always performed even after a hit.

Figure 3-3. State Diagram of SDRAM Controller



(1) Read cycle

The read cycle supports the burst read cycle of burst length = 4. After burst read cycle, miss must always be performed (when controlling precharge operation in SDRAM).

(2) Write cycle

The write cycle supports burst write cycle with burst length = 4 in the same way as read cycle.

(3) Refresh cycle

The refresh cycle supports CBR (auto) refresh cycle.
Refresh cycle is controlled once each time a certain period elapses, and then miss is performed.
While SDRAM performs refresh cycle, the bus master performs write operation to the address in the refresh space defined in the memory space to be described later. The refresh request signals are generated utilizing the change of address signal that occurs when an address in the refresh space is selected.
The data written during this period is defined as invalid. Control must be made with software never to read out this data.

(4) Mode register set cycle

The mode register set cycle performs controls similar to those in the refresh cycle.

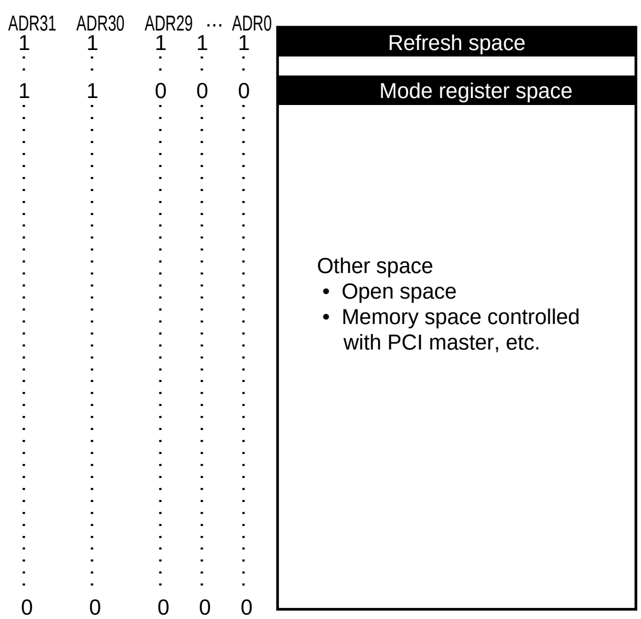
3.2.3 Memory space

Write cycle, refresh cycle, and mode register set cycle are distinguished according to the memory space. Controls with the memory space enables mode register set cycle to be input at any location. In addition, some SDRAM products with the same capacity but different refresh period can also be supported.

Figure 3-4 shows an example of memory space. In this example, the memory space is allocated as follows.

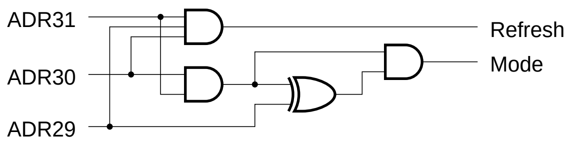
- ADR0 to ADR31 are 1: Refresh space
- ADR0 to ADR29 are 0, ADR30 and ADR31 are 1: Mode register space

Figure 3-4. Example of Memory Space of PCI Controller



Input the address signal (ADR31 to ADR29) defined in Figure 3-4 to the circuit in Figure 3-5 to generate the refresh signal and mode signal. When the refresh space is selected, the refresh signal becomes high, and when the mode register space is selected, the mode signal becomes high.

Figure 3-5. Select Signals of Refresh and Mode Register Cycle



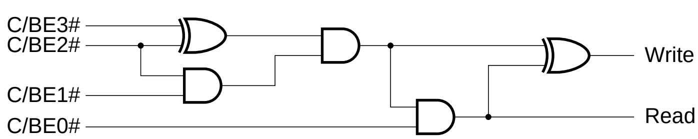
ADR31	ADR30	ADR29	Refresh	Mode
1	1	1	1	0
1	1	0	0	1
1	0	1	0	0
1	0	0	0	0
0	1	1	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	0	0

Input C/BE3#, C/BE2#, C/BE1#, and C/BE0# signals to the circuit in Figure 3-6 to generate the read signal and write signal.

With the memory read command of the PCI bus, the read signal becomes high. With the memory write command of the PCI bus, the write signal becomes high.

The control of read/write cycle uses the logic level of C/BE# signal output from the PCI bus to generate read and write control signals. The selection of memory space according to read/write cycle, therefore, is not required. In this case, two SDRAMs perform identical operations.

Figure 3-6. Select Signals of Read/Write Cycle



C/BE3#	C/BE2#	C/BE1#	C/BE0#	Read	Write
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Table 3-1 summarizes Figures 3-5 and 3-6.

Table 3-1. Select Signals of Each Operation Mode

Operation Mode of SDRAM	Read	Write	Refresh	Mode
Read cycle	1	0	0	0
Write cycle	0	1	0	0
Refresh cycle	0	1	1	0
Mode register set cycle	0	1	0	1

- Read cycle
  Only read signal always becomes high.
- Write cycle
  Only write signal always becomes high.
- Refresh cycle
  Refresh signal and write signal always become high.
- Mode register set cycle
  Mode signal and write signal always become high.

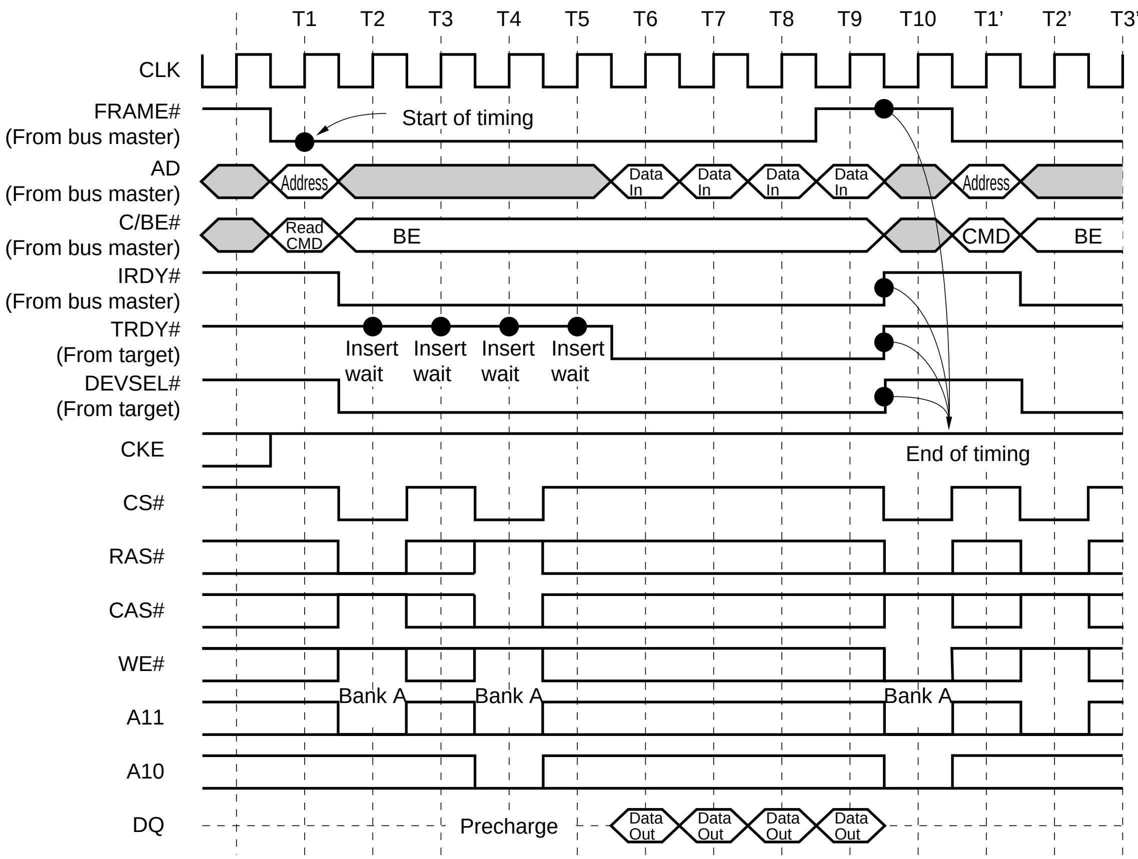
3.3 Timings between PCI Controller and SDRAM

This section explains the timing between the PCI controller and SDRAM.

3.3.1 Read cycle of SDRAM

Burst read cycle with burst length = 4 is used for read cycle.

Figure 3-7. Read Cycle between PCI Controller and SDRAM



The bus master inputs read command at T1 and waits data input from SDRAM.

An active command, a read command, a precharge command, and an active command that indicates the next operation are input to SDRAM at T2, T4, T10, and T2', respectively.

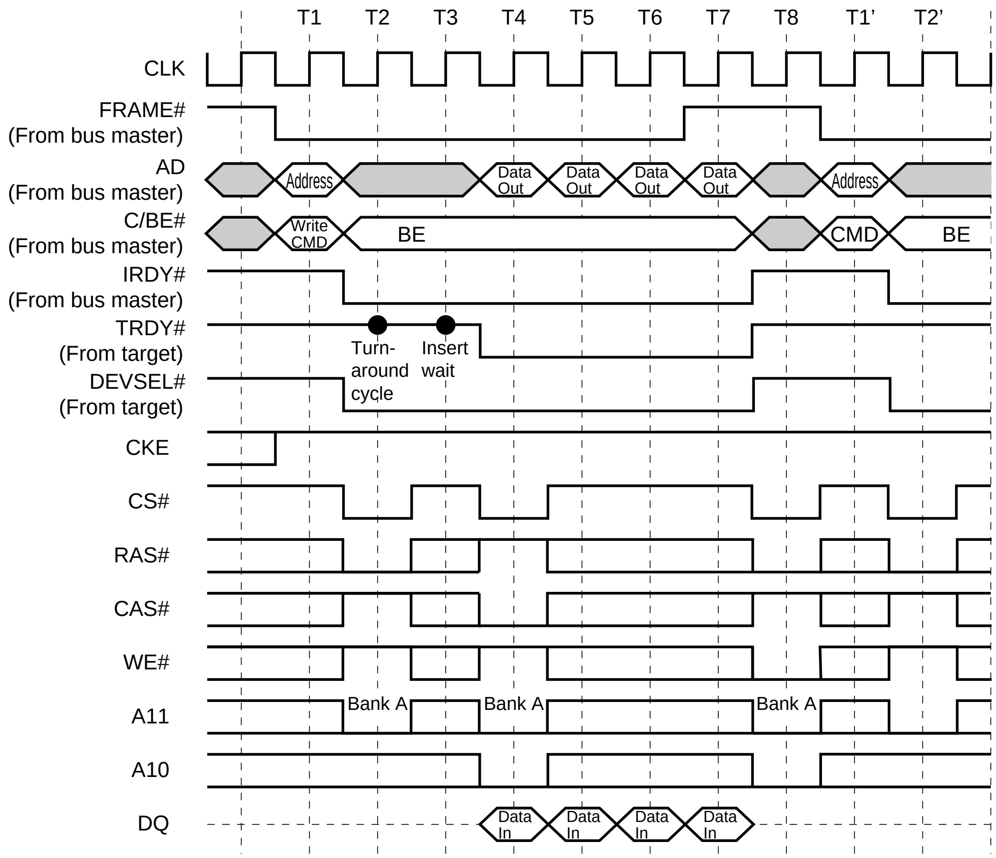
A total of four waits are input at T2, T3, T4, and T5.

Although only bank A is accessed in Figure 3-7, bank B also can be accessed. Although this circuit uses precharge, auto precharge can also be used without affecting the operation.

3.3.2 Write cycle of SDRAM

Burst write cycle with burst length = 4 is used for write cycle as well as for read cycle.

Figure 3-8. Write Cycle between PCI Controller and SDRAM



The bus master inputs a write command at T1 and declares output of data to SDRAM.

An active command, a write command, a precharge command, and an active command that indicates the next operation are input to SDRAM at T2, T4, T8, and T2', respectively.

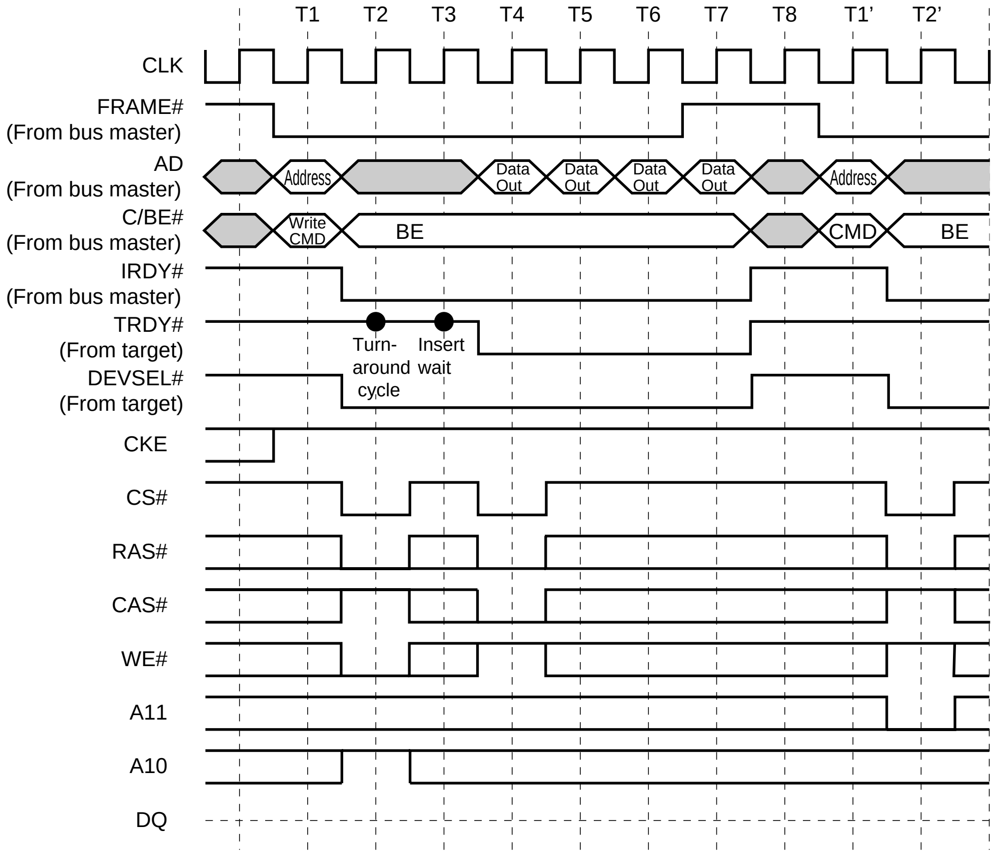
A turn-around cycle is inserted at T2, and a wait is inserted at T3.

Although only bank A is accessed in Figure 3-8, bank B also can be accessed. Although this circuit uses precharge, auto precharge can also be used without affecting the operation.

3.3.3 CBR (auto) refresh cycle of SDRAM

CBR (auto) refresh cycle is used for refresh cycle, and it is multiplexed with the timing of the write cycle.

Figure 3-9. CBR (Auto) Refresh Cycle between PCI Controller and SDRAM



The bus master inputs a write command at T1 and declares output of data to memory.

An all bank precharge command, CBR (auto) refresh command, and an active command that indicates the next operation are input to SDRAM at T2, T4, and T2’, respectively. In this case, the bus master outputs data, but SDRAM does not accept the data. For the reason, refer to **3.2.2 State diagram of SDRAM controller**.

One turn-around cycle and one wait are inserted at T2 and T3, respectively. Turn-around cycle, however, is not essentially required in CBR (auto) refresh cycle. This is because CBR (auto) refresh cycle is multiplexed with the write cycle.

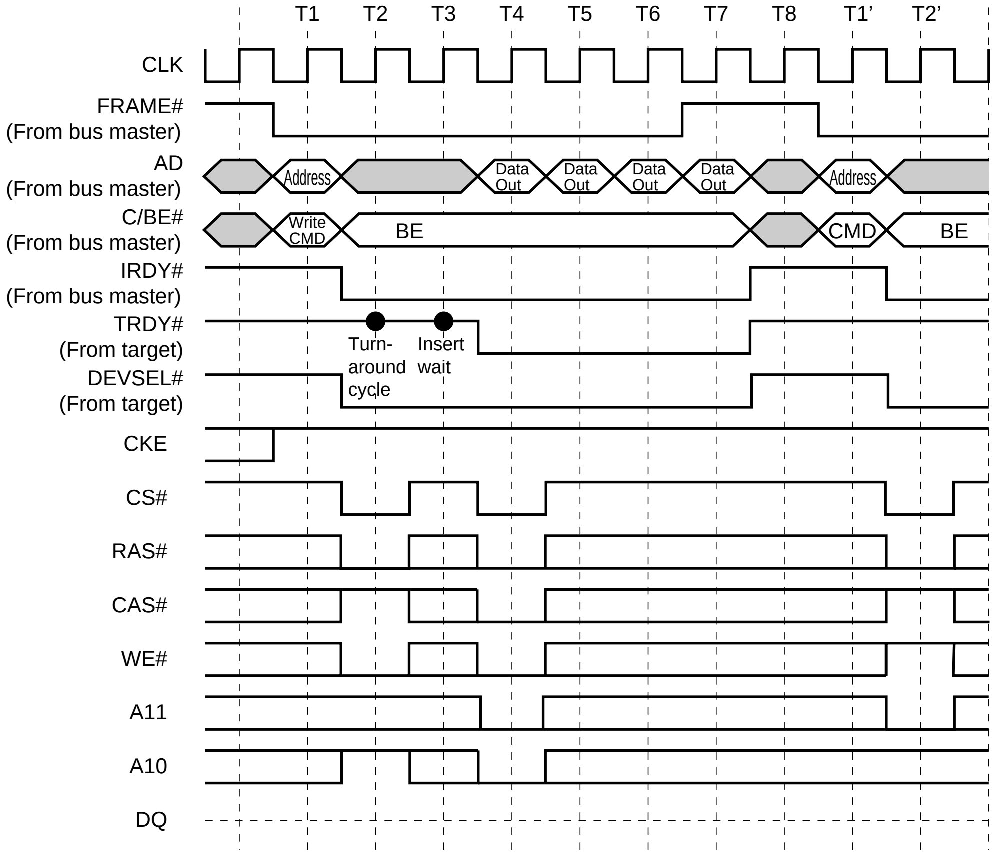
This method has the advantage that the refresh cycle generation circuit can be simplified. On the other hand, it has the disadvantage that it takes time to control the refresh cycle because of the insertion of unnecessary waits. In terms of the overall data transfer time including the bus master and SDRAM, higher speed can be realized by not multiplexing CBR (auto) refresh cycle and write cycle because of less control over refresh cycle.

In this case, however, the method multiplexing CBR (auto) refresh cycle and write cycle is used to simplify the circuit rather than generating its own refresh cycle timings.

3.3.4 Mode register set cycle of SDRAM

Mode register set cycle is multiplexed with the timing of the write cycle.

Figure 3-10. Mode Register Set Cycle between PCI Controller and SDRAM



The bus master inputs a write command at T1 and declares output of data to SDRAM.

An all bank precharge command, a mode register set command, and an active command that indicates the next operation are input to SDRAM at T2, T4, and T2', respectively. In this case, the bus master outputs data, but SDRAM does not accept the data. For the reason, refer to **3.2.2 State diagram of SDRAM controller**.

One turn-around cycle and one wait are inserted at T2 and T3, respectively. Turn-around cycle, however, is not essentially required in mode register set command as well as in refresh cycle. The mode register set cycle is the same as the write cycle because the mode register set cycle is multiplexed with the write cycle. The reason for this is the same as in the case of CBR (auto) refresh cycle.

3.3.5 Initialization cycle

As shown in **1.4.3 Write command and precharge**, SDRAM requires initialization when the power is turned on. Initialization of SDRAM can be controlled with software.

The following shows the procedure of initialization.

1. Do not access SDRAM until 100 μs from power on.
2. Select refresh space in the memory space (refer to **Figure 3-4**) twice consecutively, and then, execute CBR (auto) refresh cycle twice consecutively.

Execution of these two operations enables generation of initializing cycle easily with software, so that initialization of SDRAM can be executed without an initialization signal generation circuit.

3.4 Examples of Connection Circuit between SDRAM and PCI Bus

This section introduces examples of connection circuits between SDRAM and the PCI bus using circuit diagrams and timing charts of the logic verification result.

The timing charts of the logic verification result have taken propagation delay in consideration. "#" and "-" postfixed to pin names and signal names in the diagram indicate active low.

The circuit examples show the concept of operation and not actual operation. Use the description only for reference purpose upon designing.

3.4.1 Basic signals

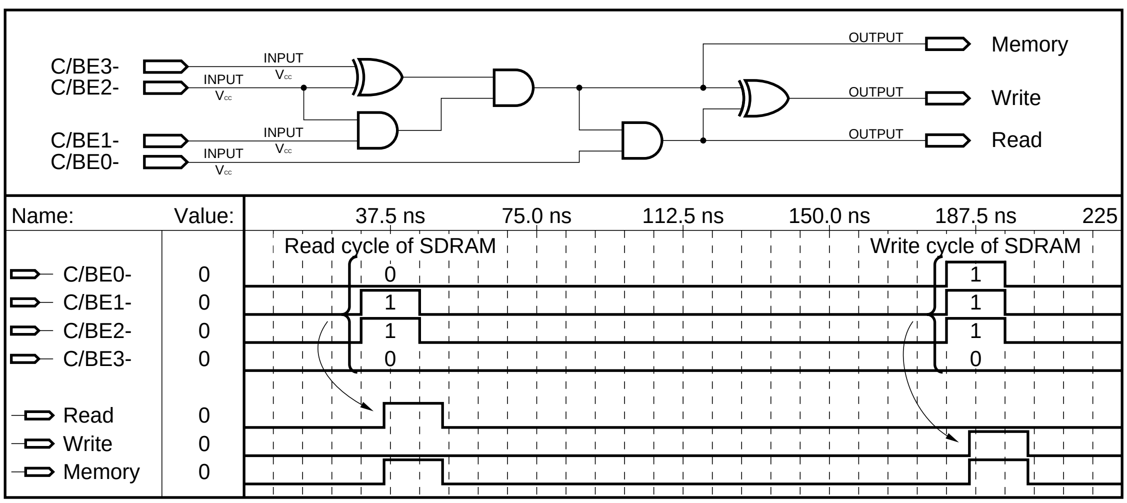
(1) SDRAM operation mode select signal

In Figure 3-11, the select signal to request the operation of each operation mode from SDRAM is generated using the circuit shown in Figure 3-6.

The memory signal becomes high only when the PCI bus command is either memory read or memory write command.

Read signal and write signal become high only when the PCI bus command is memory read and when it is memory write, respectively.

Figure 3-11. Circuit Diagram and Timings of Select Signal Generation of Each Operation Mode



(2) Basic control signal

In Figure 3-12, basic control signals are generated. These signals are referenced when generating the command to control SDRAM.

AAA signal : Becomes low at the following rising edge of CLK signal when FRAME# signal is high, and becomes high when FRAME# signal is low.

BBB signal : 1/2 frequency division signal of CLK signal started from low when FRAME# and AA signals are low.

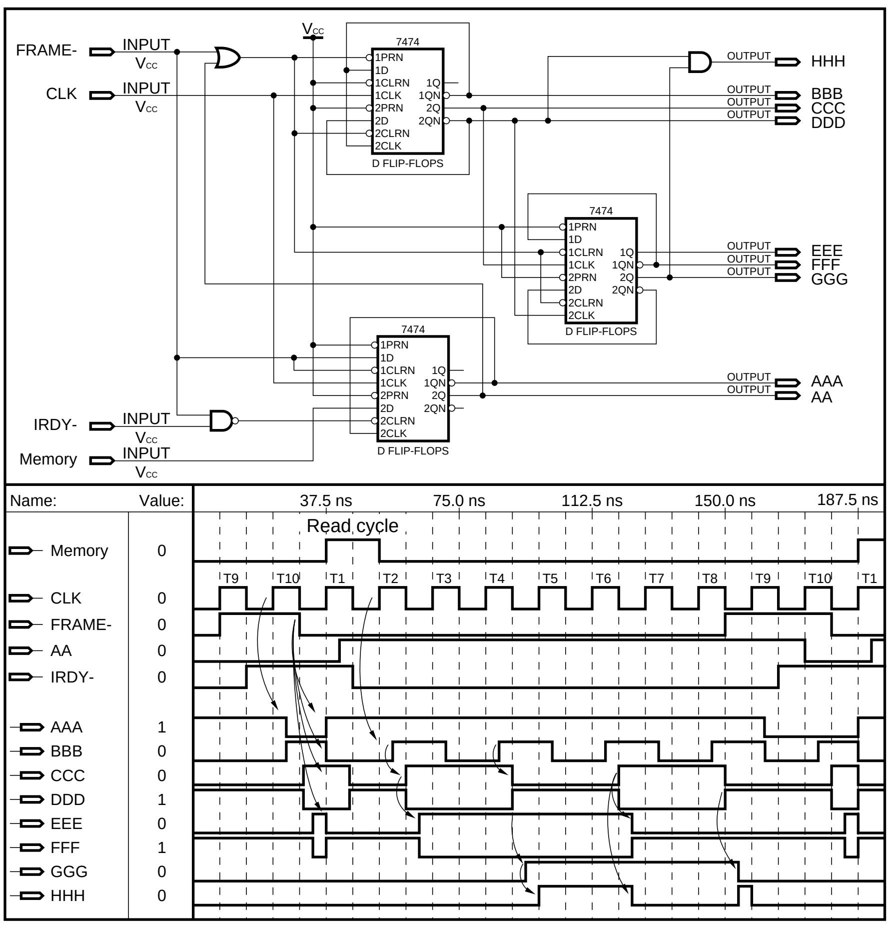
CCC signal : 1/2 frequency division signal of BBB signal (1/4 frequency division of CLK). DDD signal is the inverted signal of CCC signal.

EEE signal : 1/2 frequency division signal of CCC signal (1/8 frequency division of CLK). FFF signal is the inverted signal of EEE signal.

GGG signal : Becomes low when FRAME# signal and AA signal are both low. When FRAME# signal and AA signal are both high, GGG signal changes at the rising edge of DDD signal.

HHH signal : Signal that employs AND logic of DDD signal and GGG signal.

Figure 3-12. Circuit Diagram and Timings of Basic Control Signal Generation



(3) Command select signal

In Figure 3-13, the basic select signal is generated. This is the signal to sort commands of various operation modes. This is summarized as follows.

Signal Name	Original Signal	Logic Verification Result
FMS1	Signal to latch read signal	Low only in cycles other than read cycle and memory
FMS2	Signal to latch the signal that employs AND logic of memory signal and mode signal	Low only in mode register set cycle
FMS3	Signal to latch memory signal	Low only in cycles other than memory
FMS4	Signal to employ AND logic of FMS0 signal and FMS2 signal	Low only in refresh cycle and mode register set cycle

All the FMS signals are high in conditions other than above.

Figure 3-13. Circuit Diagram of Basic Select Signal Generation

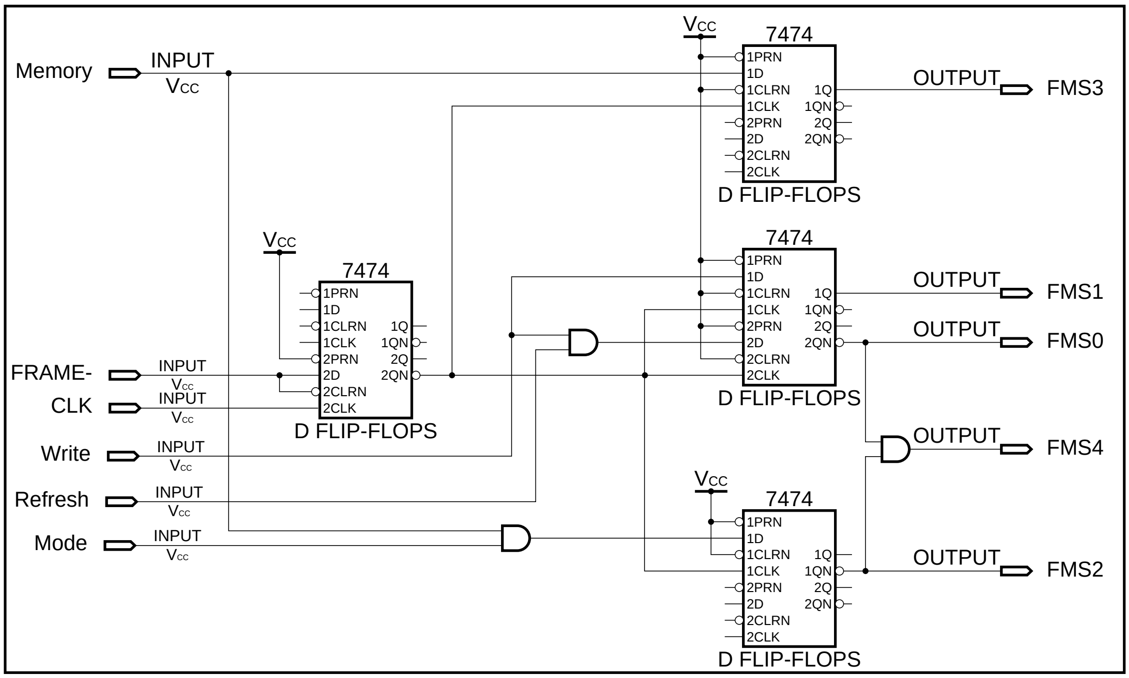
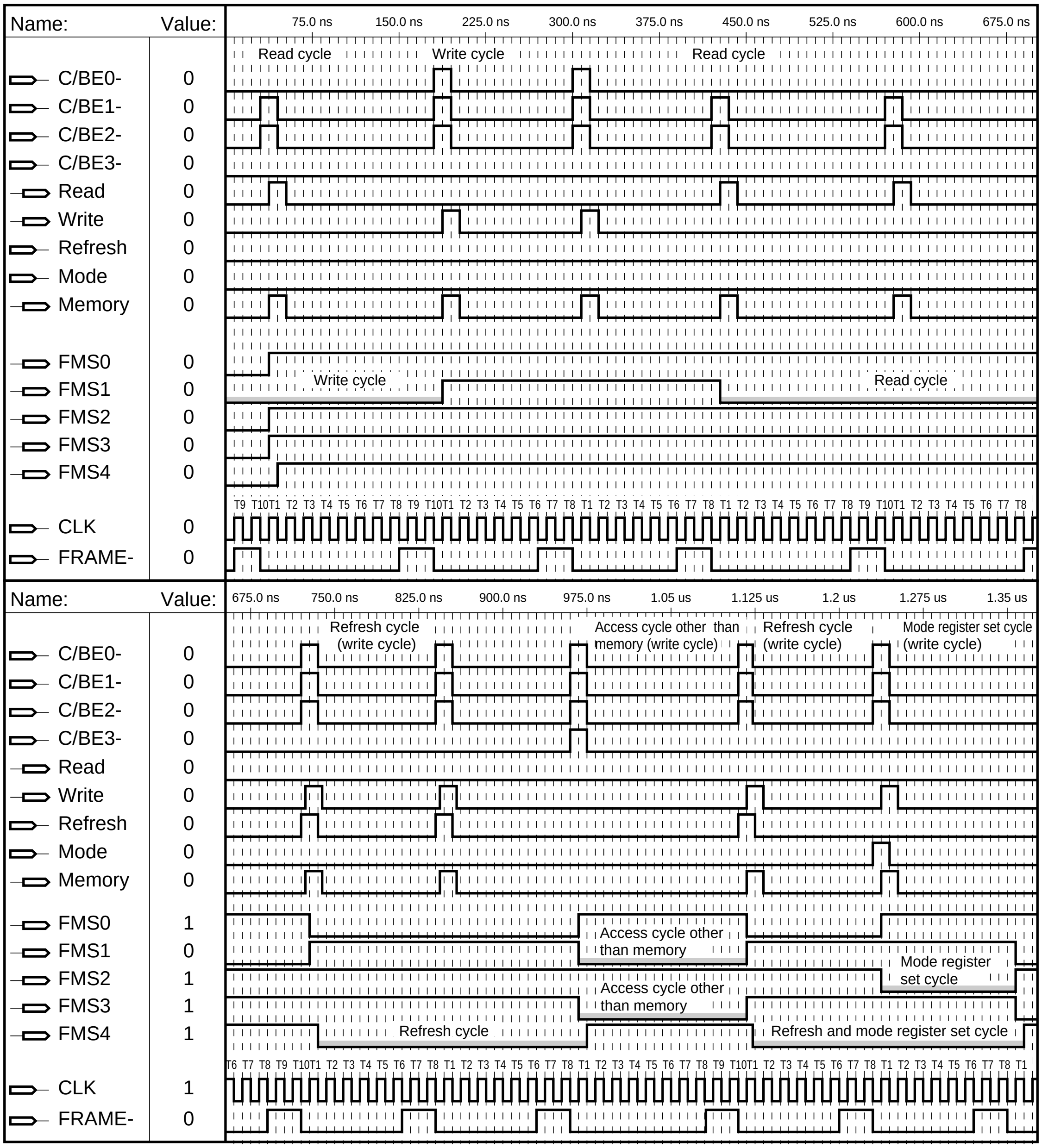


Figure 3-14. Timings of Basic Signal



3.4.2 Read cycle

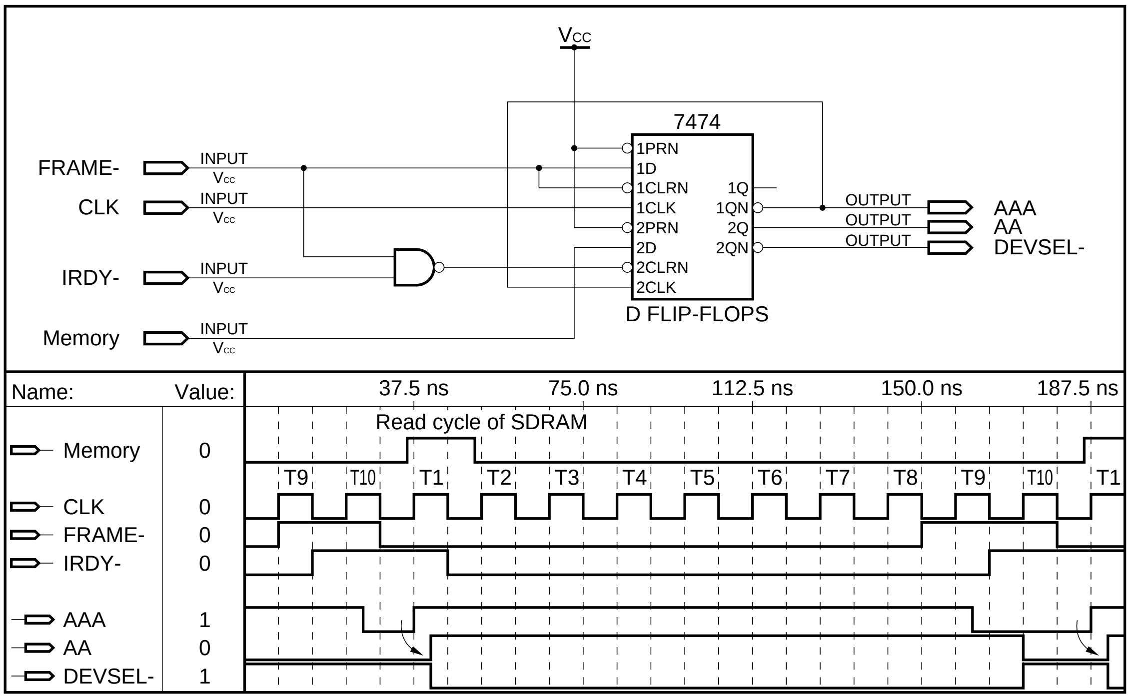
The read cycle in the circuit example supports only the burst read cycle with burst length = 4.

(1) DEVSEL# signal

In Figure 3-15, DEVSEL# signal is generated.

The DEVSEL# signal is generated by inverting AA signal. The DEVSEL# signal becomes low at the rising edge of the AAA signal only when memory signal is high, and it becomes high when the FRAME# signal and IRDY# signal are both high.

Figure 3-15. Circuit Diagram and Timings of DEVSEL# Signal Generation

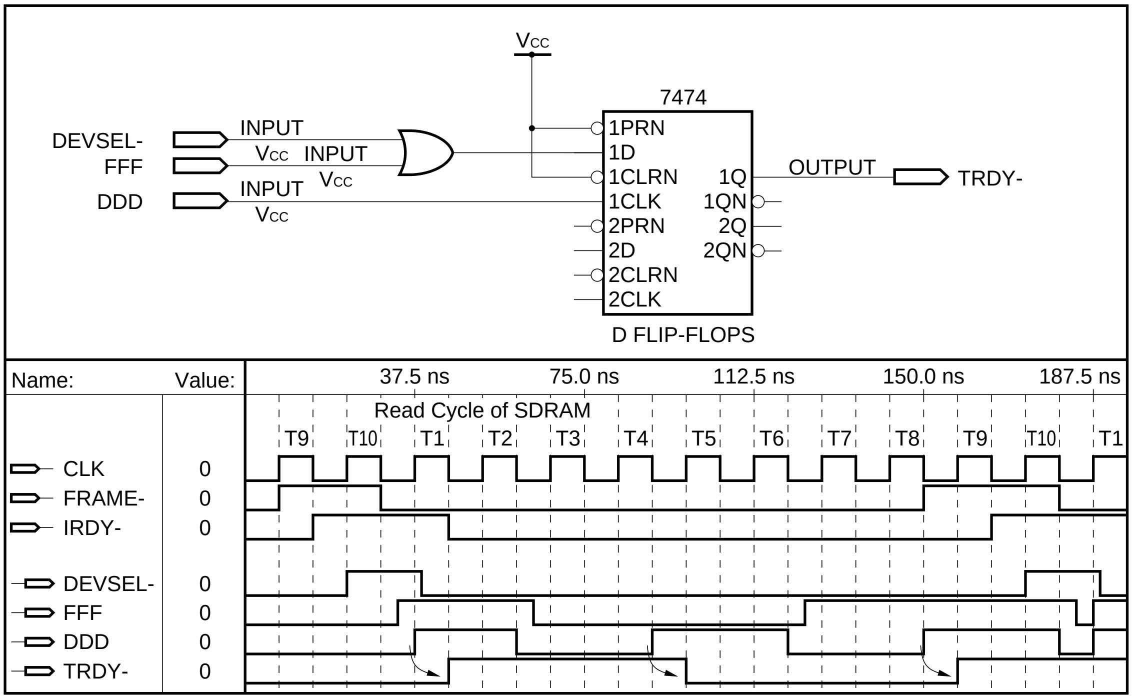


(2) TRDY# signal

In Figure 3-16, the TRDY# signal is generated.

The TRDY# signal is generated by latching the OR logic signal of the DEVSEL# signal and FFF signal at the rising edge of the DDD signal.

Figure 3-16. Circuit Diagram and Timings of TRDY# Signal Generation

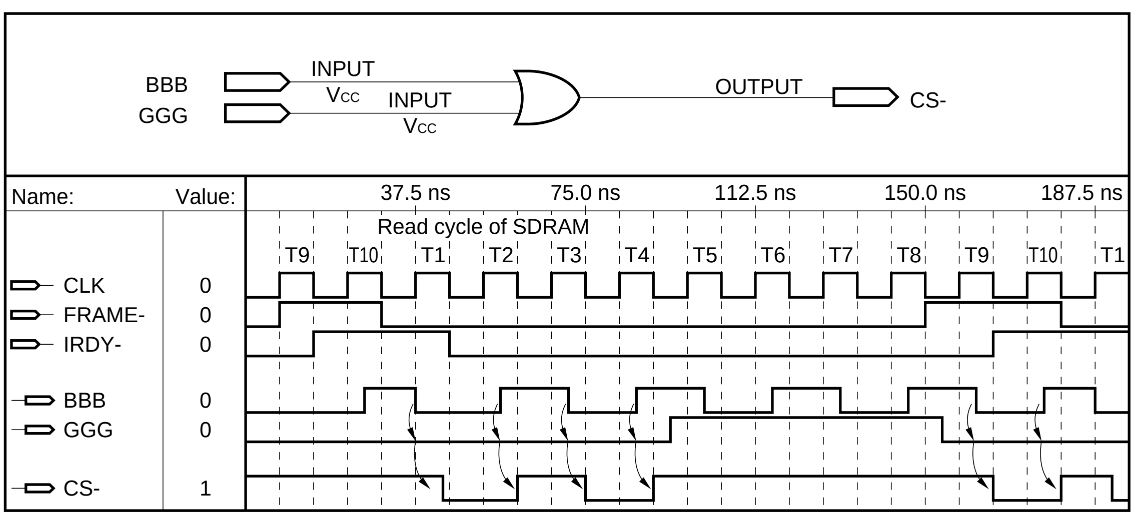


(3) CS# signal

In Figure 3-17, the CS# signal is generated.

Because BBB signal is used as the CS# signal, unnecessary clocks are deleted by employing OR logic of the BBB signal and GGG signal.

Figure 3-17. Circuit Diagram and Timings of CS# Signal Generation

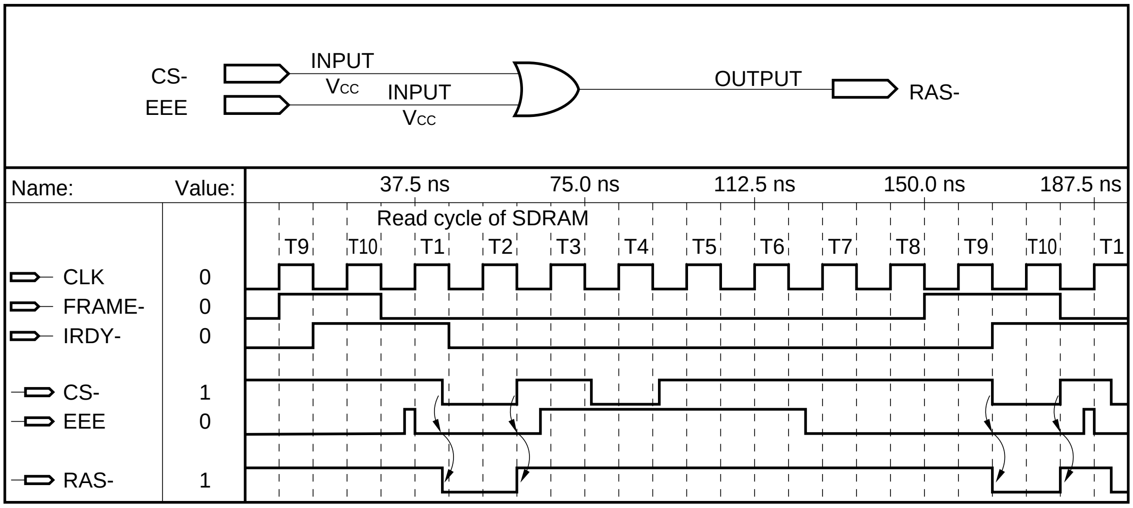


(4) RAS# signal

In Figure 3-18, the RAS# signal is generated.

Because the CS# signal is used as the RAS# signal, unnecessary clocks are deleted by employing OR logic of the CS# signal and EEE signal.

Figure 3-18. Circuit Diagram and Timings of RAS# Signal Generation

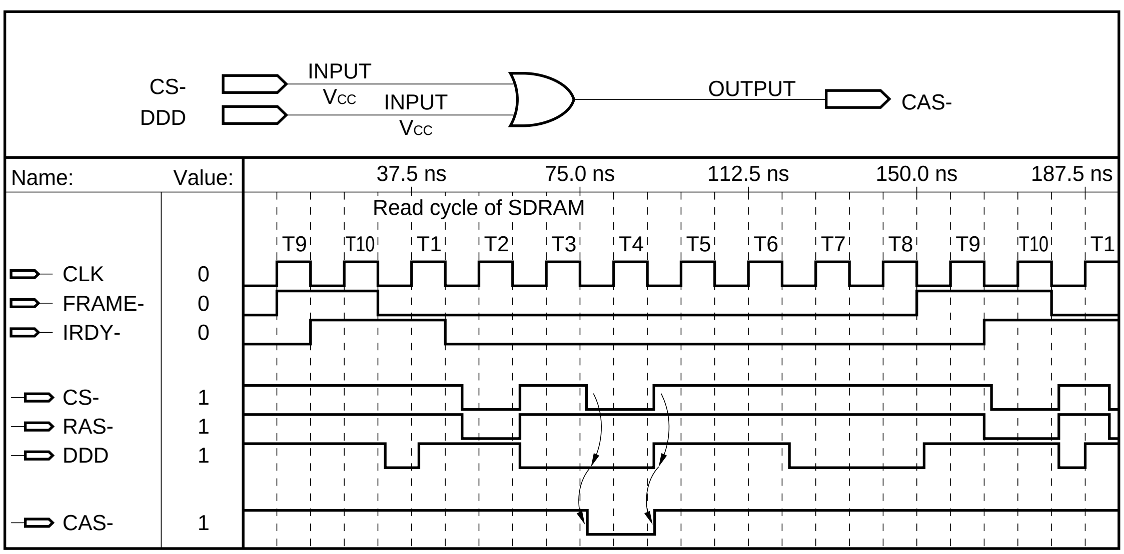


(5) CAS# signal

In Figure 3-19, CAS# signal is generated.

Because the CS# signal is used as the CAS# signal, unnecessary clocks are deleted by employing OR logic of the CS# signal and DDD signal.

Figure 3-19. Circuit Diagram and Timings of CAS# Signal Generation



(6) A11 signal and A10 signal

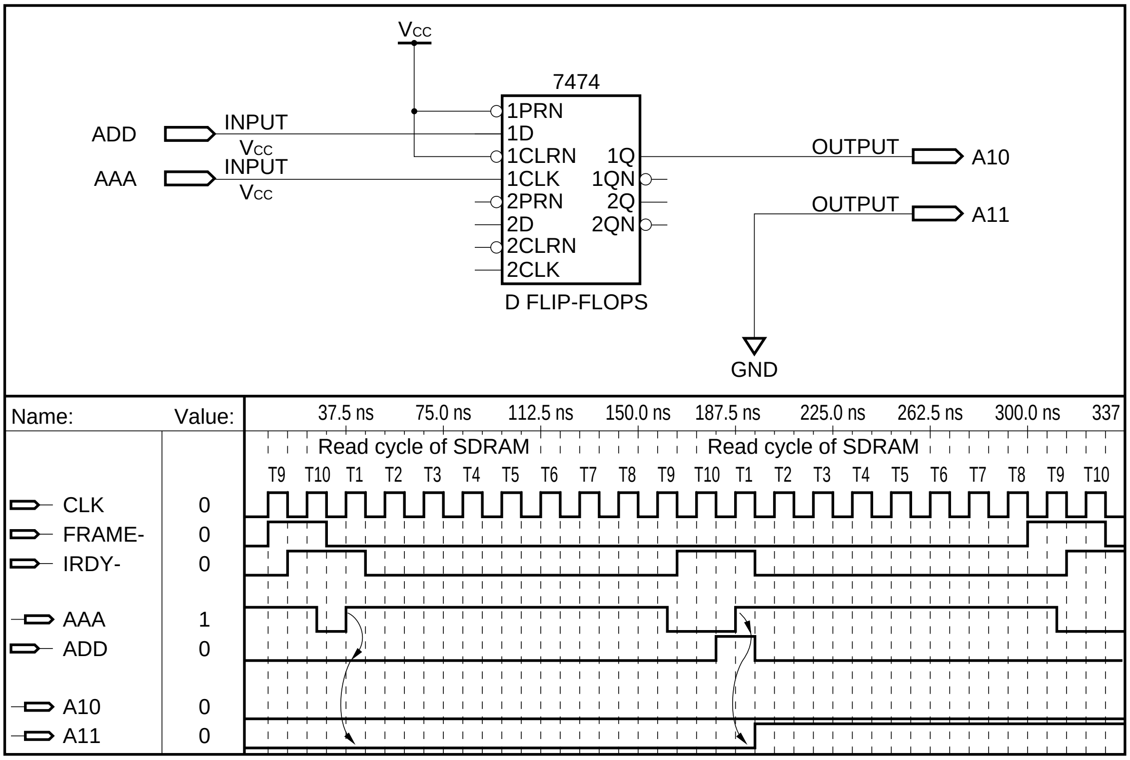
Figure 3-20 illustrates the circuit to generate the A11 signal and A10 signal.

The A11 signal is the address signal of SDRAM to control banks. The A10 signal is the address signal of SDRAM to control precharge operations.

The A11 signal is generated at the rising edge of the AAA signal by latching ADD (lower address signal output from the PCI bus controller).

Because the read/write operation in the circuit example uses only precharge, the A10 signal is always fixed to low.

Figure 3-20. Circuit Diagram and Timings of A11 Signal and A10 Signal Generation

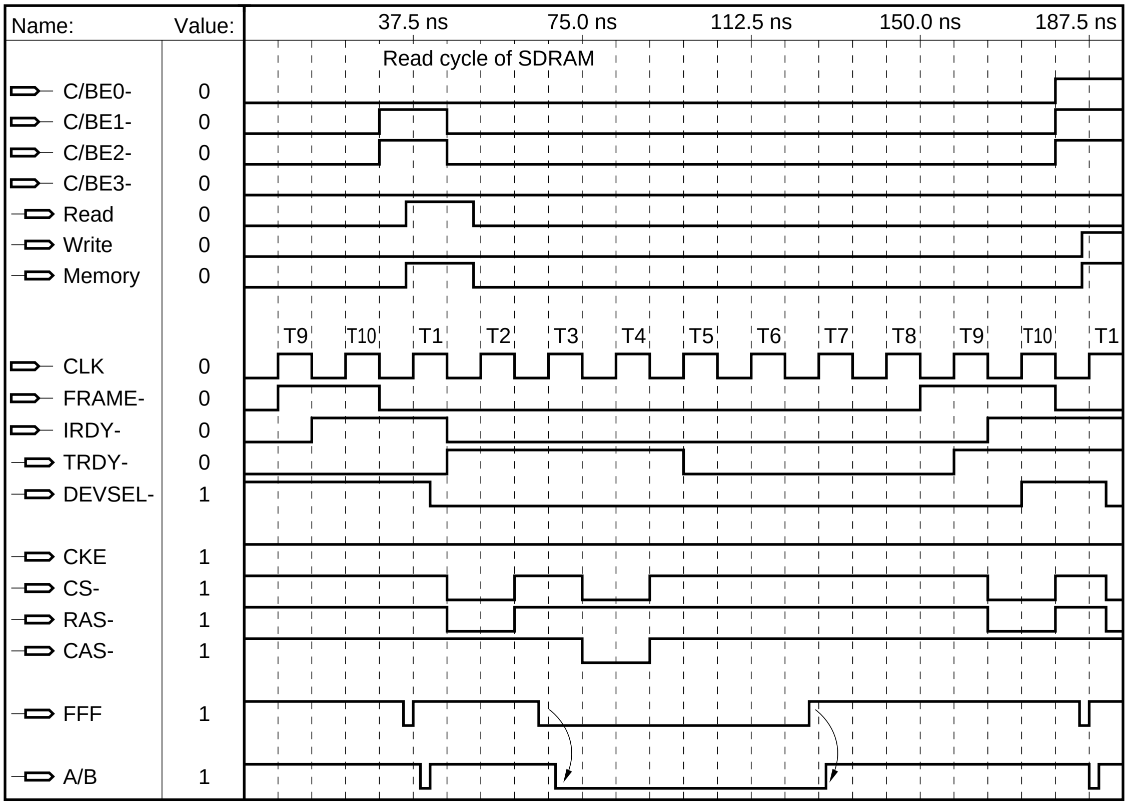


(7) Changing row/column address signal

In Figure 3-21, the A/B signal is generated. This signal is used for changing row address signal and column address signal of SDRAM.

The A/B signal must be changed between the point where row address signal is latched (T2) and the point where column address signal is latched (T4). The A/B signal uses the FFF signal as is.

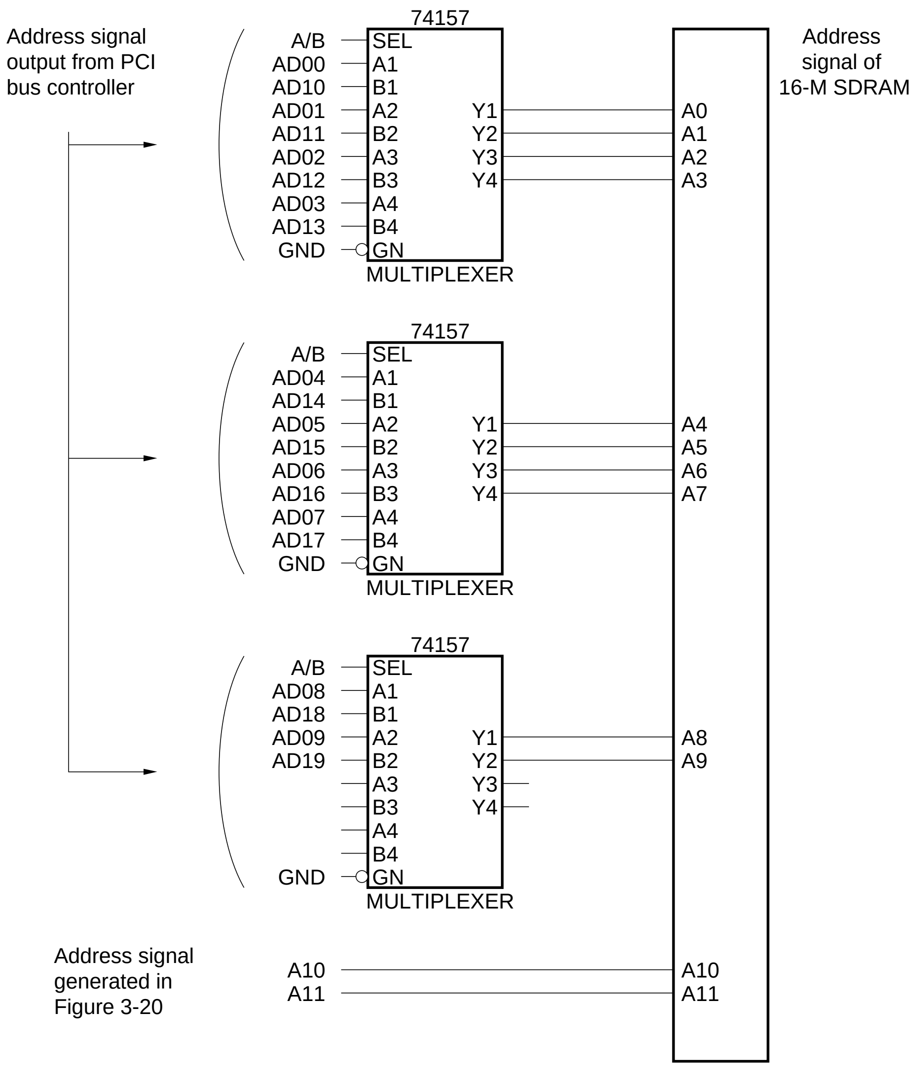
Figure 3-21. Timings of Row/Column Address Signal Changing Signal



(8) Example of address signal changing circuit using A/B signal

The A/B signal is connected to the SEL pin of 74157 as shown in Figure 3-22. The address signal output from the PCI bus controller is changed to row/column address signal and input to SDRAM.

Figure 3-22. Example of Address Signal Changing Circuit

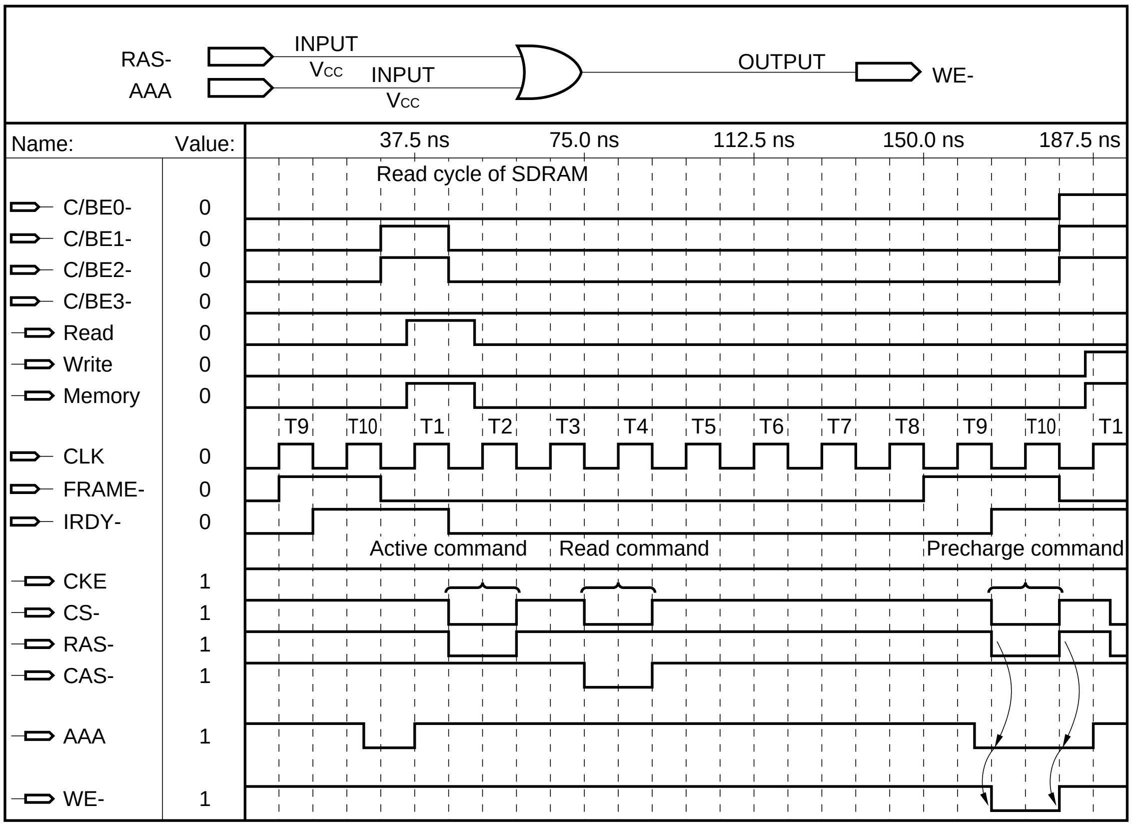


(9) WE# signal

In Figure 3-23, the WE# signal is generated.

Because the RAS# signal is used as the WE# signal, unnecessary clocks are deleted by employing OR logic of the RAS# signal and AAA signal.

Figure 3-23. Circuit Diagram and Timings of WE# Signal Generation

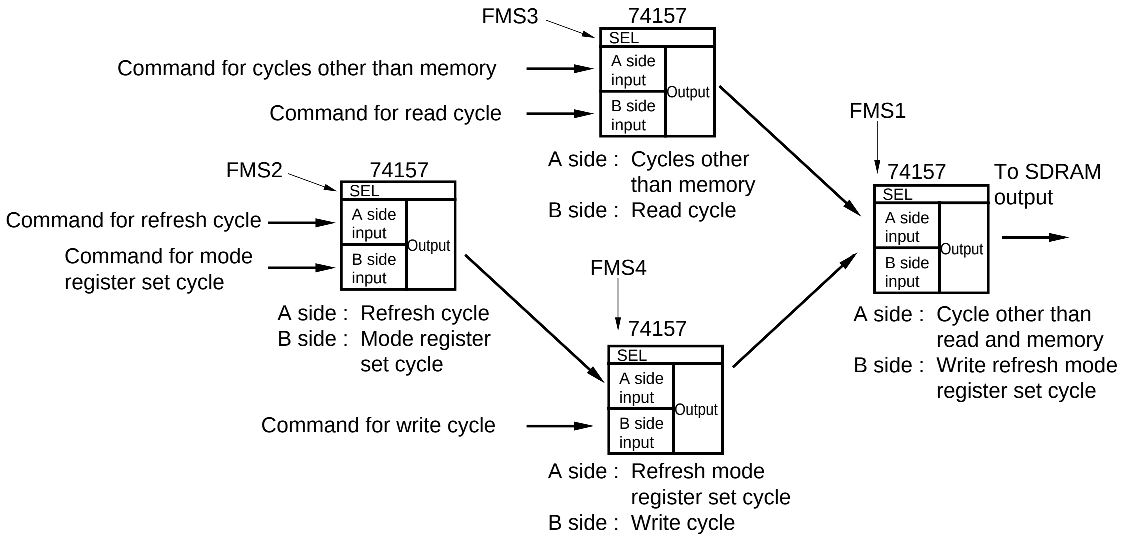


3.4.3 Changing control signal

In Figure 3-24, commands corresponding to various operation modes are changed using the FMS1, 2, 3, and 4 signals.

In the circuit example, read, write, refresh, mode register set, and accesses other than memory are controlled. The command for these operation modes are individually generated, changed according to the corresponding operation mode, and input from SDRAM controller to SDRAM.

Figure 3-24. Example of Control Signal Changing Circuit



3.4.4 Write cycle

The write cycle in the circuit example supports only the burst write cycle with burst length = 4.

Figure 3-25 shows the circuit diagram of burst write cycle control. In this diagram, the A side of 74157 is the circuit to control the command for read cycle, and the B side is the circuit to control the command for write cycle.

The CS# signal, RAS# signal, CAS# signal, and WE# signal generate commands in write cycle as OR logic of the BBB signal and HHH signal, OR logic of the CS# signal and EEE signal, OR logic of the CS# signal and FFF signal, and AND logic of the CAS# signal and WE# signal in read cycle, respectively.

Figure 3-25. Circuit Diagram of Burst Write Cycle Control Generation

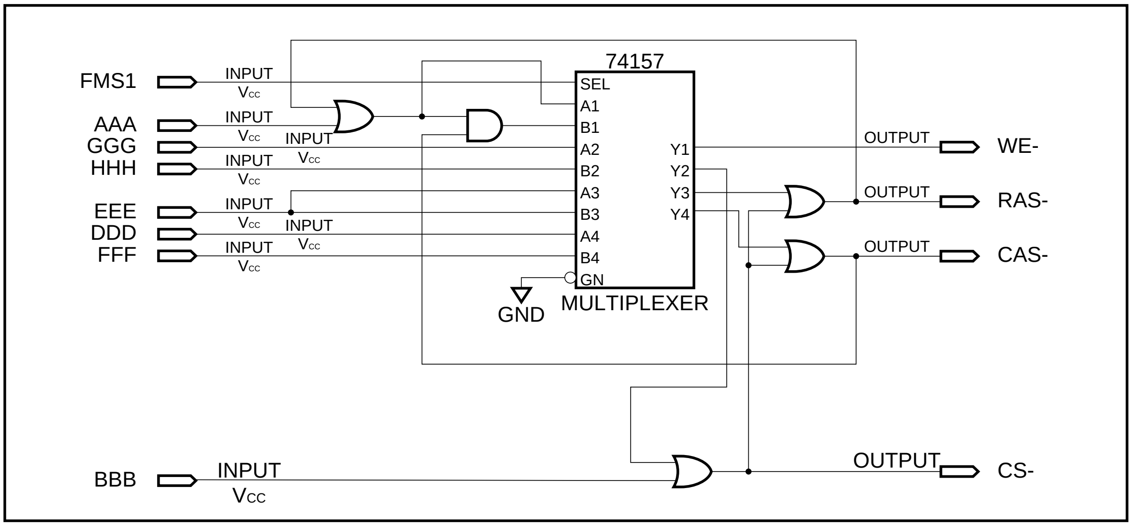
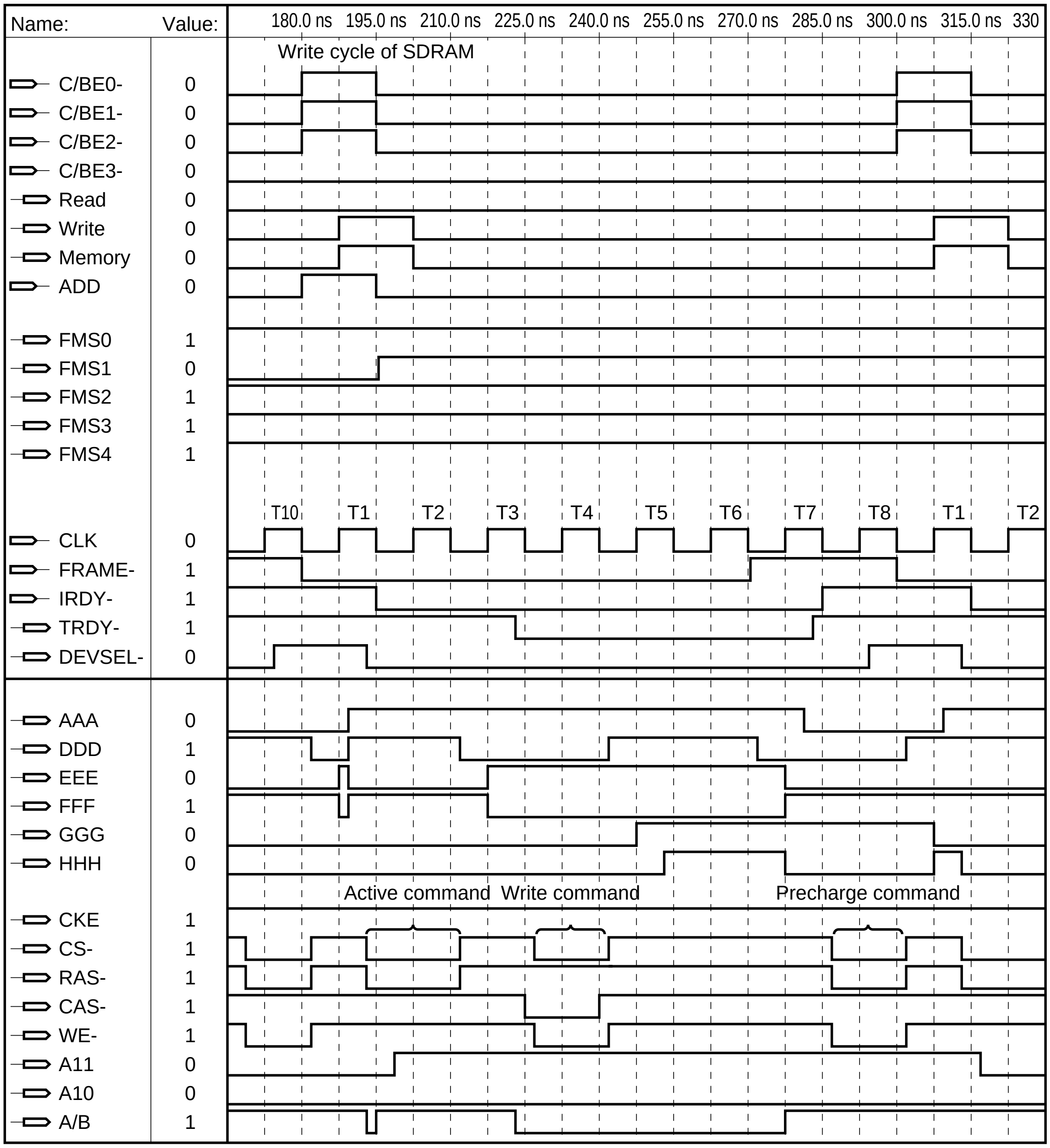


Figure 3-26. Timings of Burst Write Cycle Control



3.4.5 Refresh cycle

The refresh cycle in the circuit example supports only the CBR (auto) refresh cycle.

Figure 3-27 shows the circuit diagram of CBR (auto) refresh cycle control. In this diagram, the A side of 74157 is the circuit to control the command for refresh cycle, and the B side is the circuit to control the command for write cycle.

The commands for refresh cycle are generated as follows:

- CS# signal : OR logic of the signal that employs OR logic of the GGG signal and HHH signal and the BBB signal
- RAS# signal : OR logic of the signal that employs OR logic of the GGG signal and HHH signal and the CS# signal
- CAS# signal : OR logic of the CS# signal and FFF signal
- WE# signal : OR logic of the GGG signal and HHH signal

Figure 3-27. Circuit Diagram of CBR (Auto) Refresh Cycle Control Generation

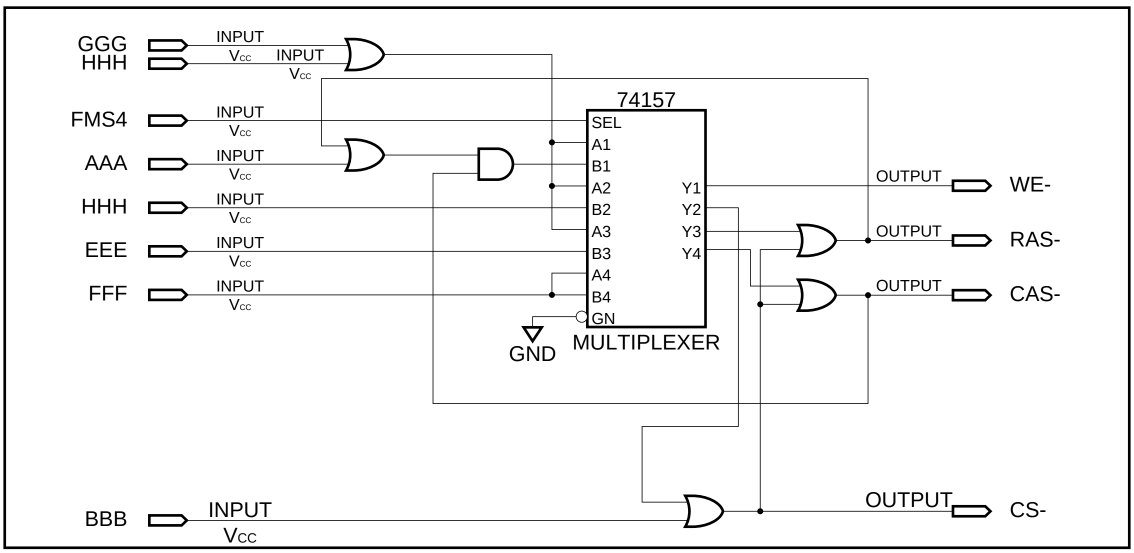
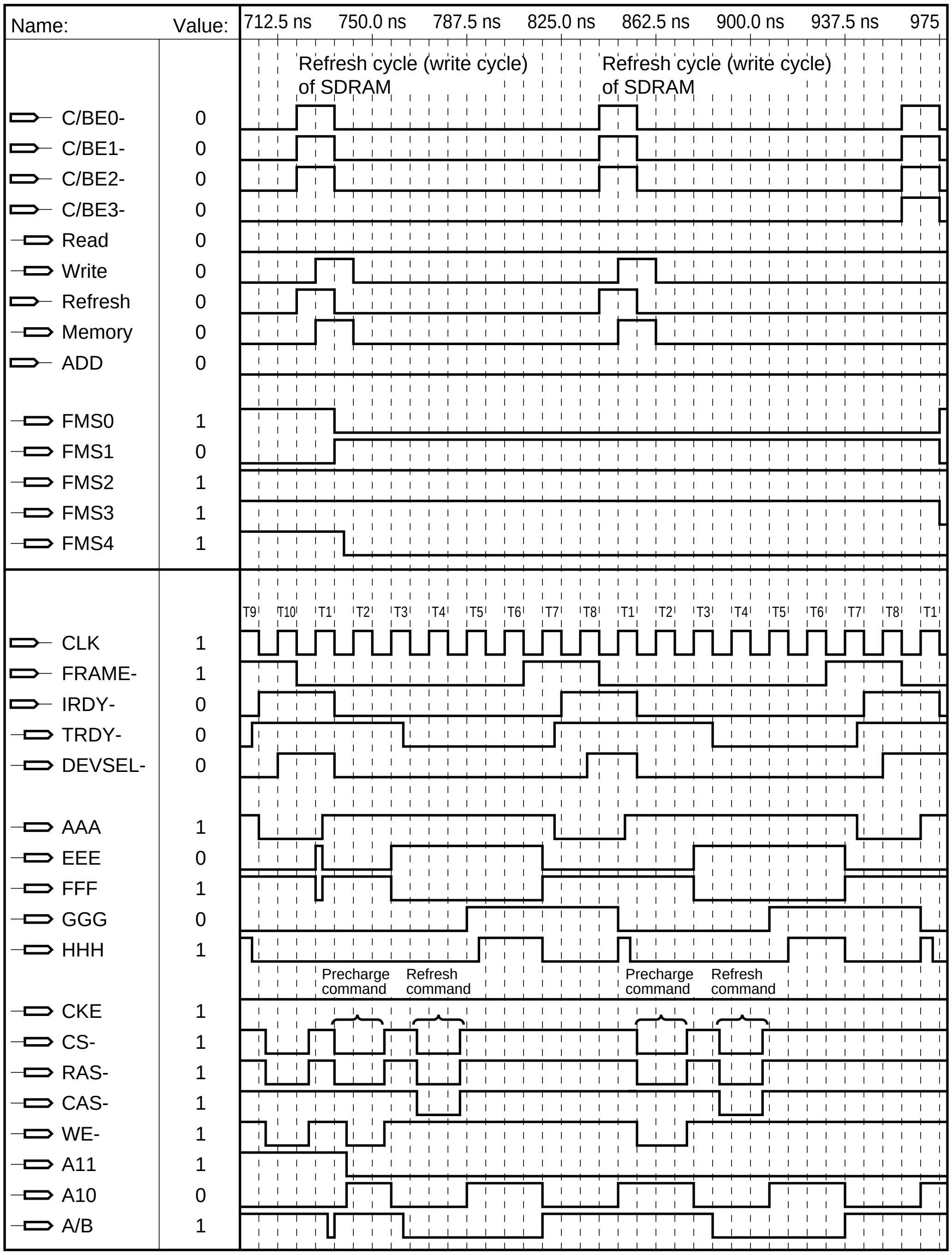


Figure 3-28. Timings of CBR (Auto) Refresh Cycle Control



3.4.6 Mode register set cycle

Figure 3-29 shows the circuit diagram of the mode register set cycle control. In this diagram, the A side of 74157 is the circuit to control the command for refresh cycle, and the B side is the circuit to control the command for mode register set cycle.

The CS# signal, RAS# signal, and CAS# signal in the mode register set cycle use the same circuit as in refresh cycle. The WE# signal generates the command in the mode register set cycle as OR logic of the CS# signal and EEE signal.

Figure 3-29. Circuit Diagram of Mode Register Set Cycle Control Generation

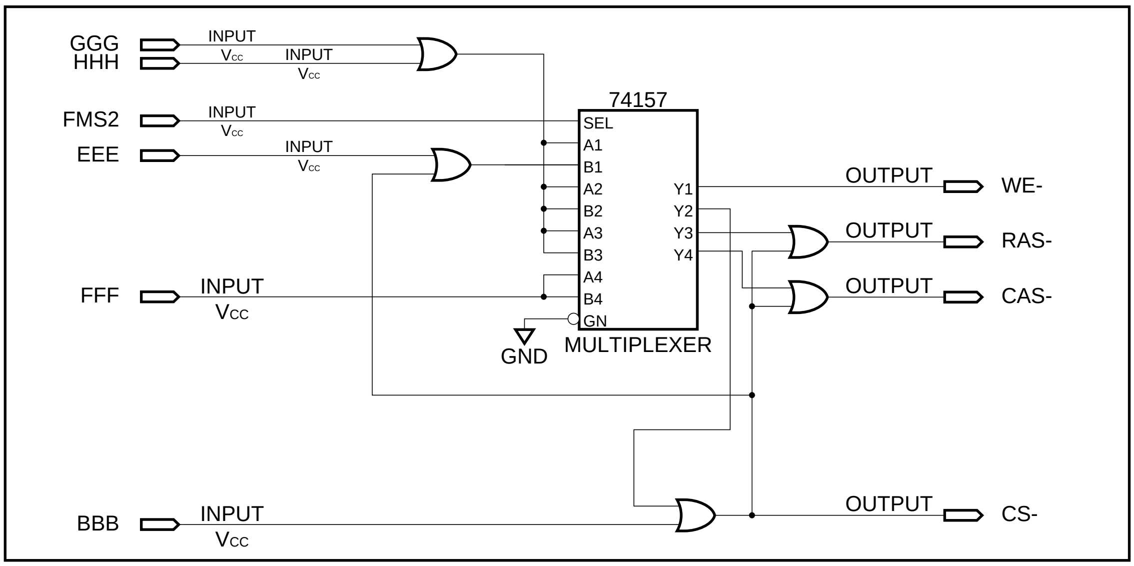
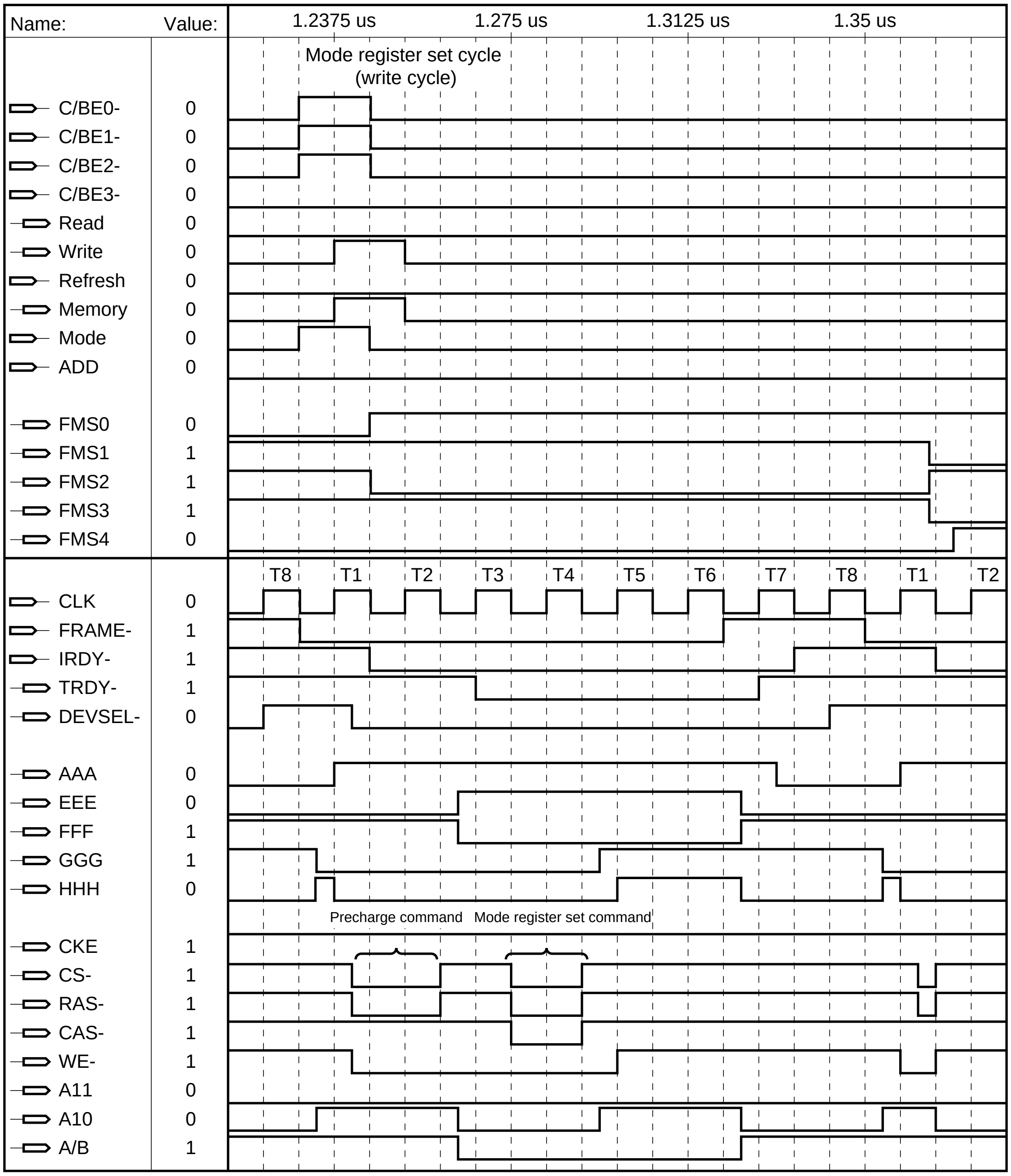


Figure 3-30. Timings of Mode Register Set Cycle Control



3.4.7 Other cycles

This section excerpts the most important of the handling of SDRAM control signals when PCI bus controller accesses devices other than memory and the operations of SDRAM that have not been explained in the preceding sections and explains their control concept.

(1) Handling of SDRAM control signal when the PCI bus controller accesses devices other than memory

Figure 3-31 shows the diagram of the circuit to control device access cycles other than memory. In this diagram, the A side of 74157 is the circuit to control the command for access cycles other than memory, and the B side is the circuit to control the command for the other cycles.

When the PCI bus controller accesses devices other than memory, CS# signal becomes high. When the CS# signal becomes high, all the control signals also become high because the RAS#/CAS# signal is generated with the OR logic with the CS# signal and WE# signal is generated with the OR logic with the RAS# signal.

Figure 3-31. Circuit Diagram of Access Cycles Other than Memory

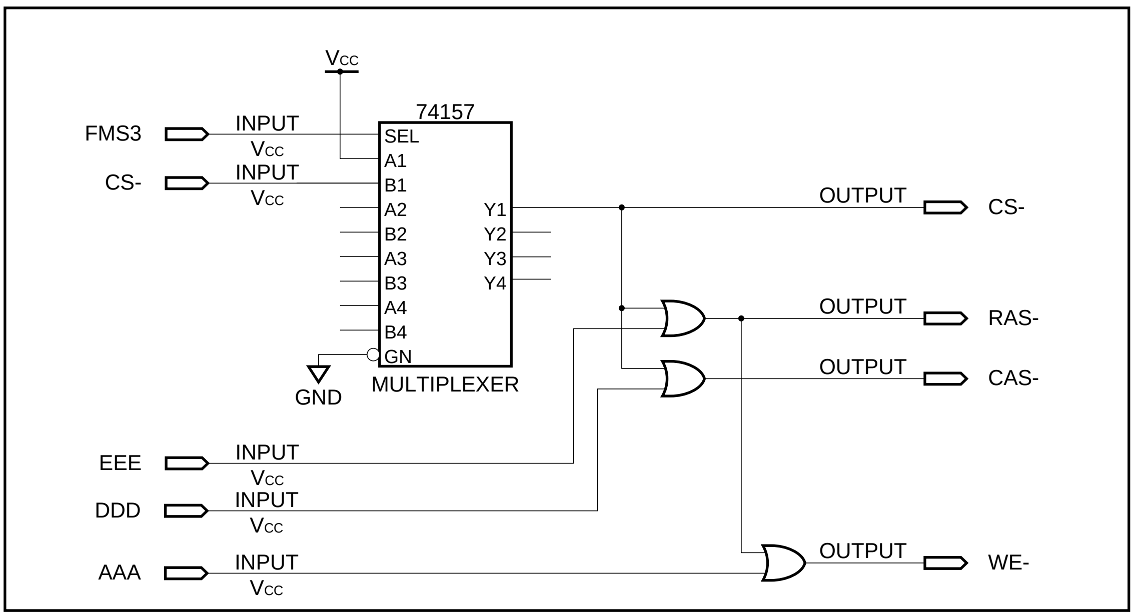
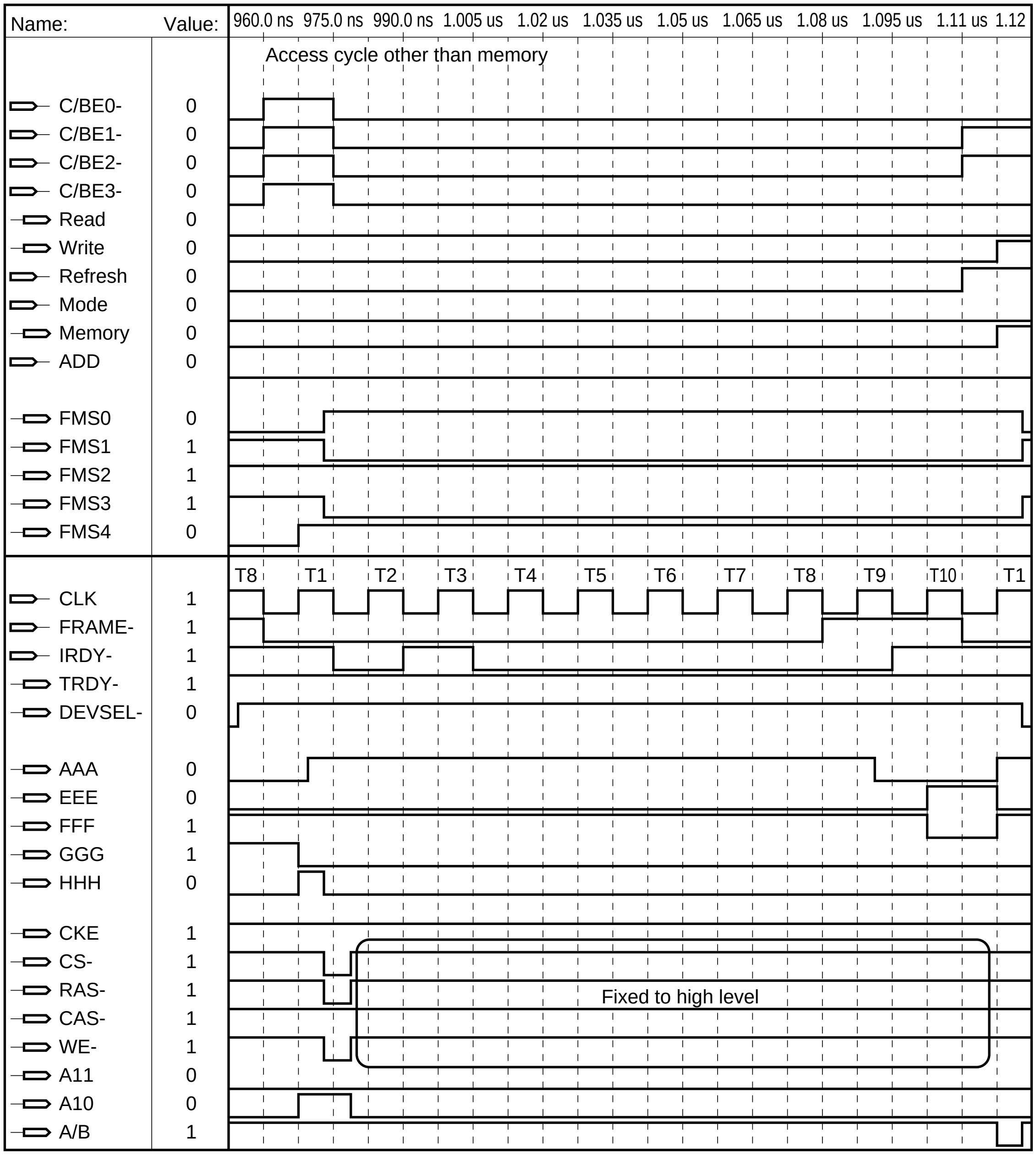


Figure 3-32. Timings of Access Cycles Other than Memory



(2) Handling of CKE signal

The CKE signal in the circuit example is fixed to high whether accessing SDRAM or not.
The controls shown below are essentially preferred for the PCI controller.

- Set the CKE signal to high when accessing SDRAM.
- Set the CKE signal to low when accessing memory other than SDRAM and I/O equipment.

This is because setting the CKE signal to low stops the internal clock of SDRAM, so that the current consumption of SDRAM can be minimized (data cannot be retained only by setting the CKE signal to low. To retain data, refresh operation is required).
In addition, if the CKE signal is set to low, SDRAM will not operate when noises are generated in control signals (CS#, RAS#, CAS#, WE#, etc.), so that malfunctions can be prevented.
In the circuit example, however, whether accessing SDRAM or not is judged by address signal from the PCI controller.
In this judging method, fixing the CKE signal to high/low one clock before the active command is too late in terms of timing. Although it is acceptable to insert a wait in order to catch the timing, inserting a wait to control the CKE signal is more disadvantageous because it delays data transfer time. The CKE signal is fixed to high in this circuit.

(3) Random access and burst access

In the circuit example, both read and write cycles are controlled with access with burst length = 4. Random access and burst access can coexist to perform controls. In this case, the following three methods are available. Method 1 is the easiest method to control.

1. Method to generate the control signal of SDRAM by extending the read random access space and the write random access space in the memory space shown in Figure 3-4 and separating random access and burst access
2. Method to use the controls of the DQM signal and CKE signal
3. Method to make the setting of the mode register again

In method 2, only one word of data can be controlled when setting burst length = 4 and mode register. This method is not recommended because the remaining three words become wait state.
In method 3, if random access and burst access are performed alternately, each setting must be made again. This method, therefore, is not recommended either.

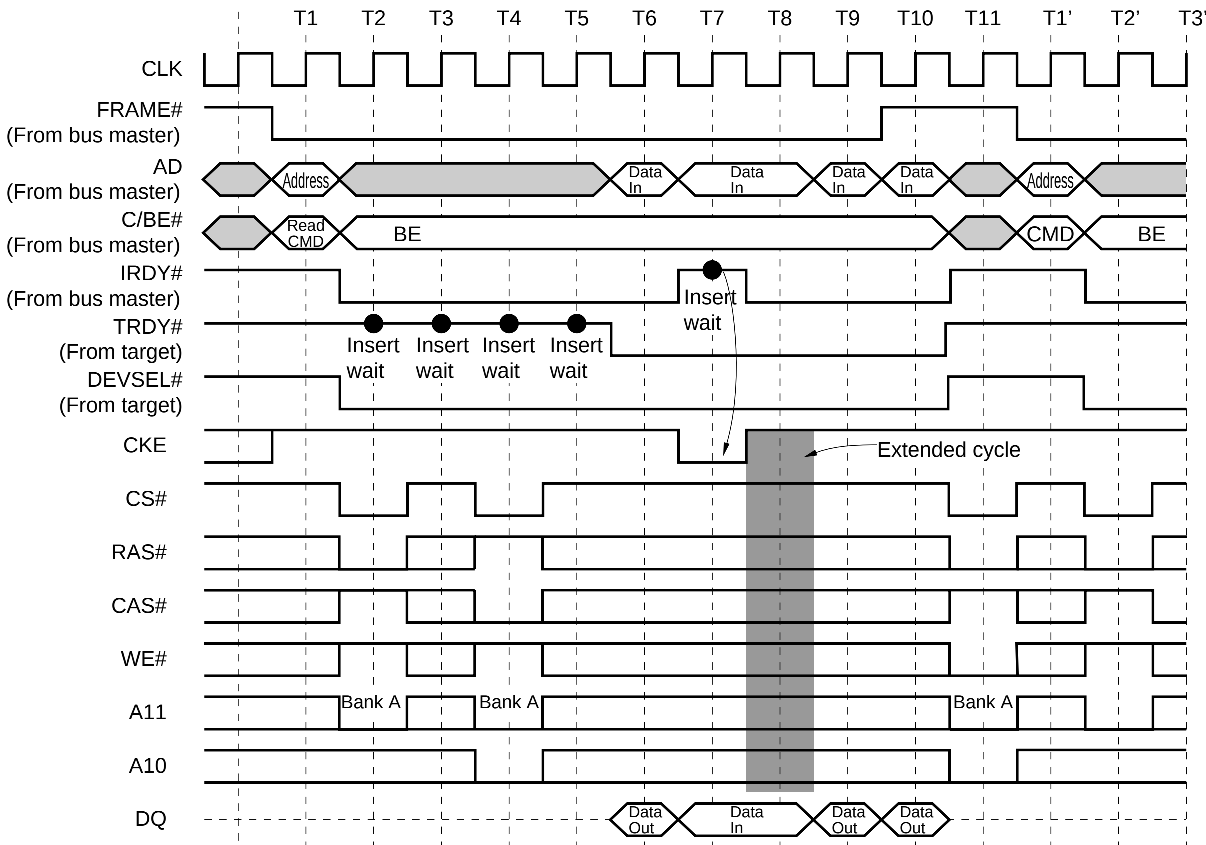
(4) When the bus master abruptly inserts a wait

In the circuit example, if the bus master inserts an unexpected wait, SDRAM will malfunction. When the bus master accesses SDRAM, therefore, the bus master side must perform controls with the wait number defined in advance.

In order to support the unexpected wait insertion from the bus master, the method to use the CKE signal is easy.

This is enabled by setting the inverted signal of the IRDY# signal as the CKE signal during data transfer cycle. Judging how many waits the bus master has inserted, the cycle currently being performed, must be extended for the number of cycles for which waits are inserted.

Figure 3-33. Waits from Bus Master



For example, in the case of read cycle in Figure 3-7, the read cycle is controlled using ten cycles. When one wait is inserted to T7 with the IRDY# signal as shown in Figure 3-33, on the other hand, extended cycles must be inserted for one cycle immediately after the wait is inserted (extended cycle must be inserted for two cycles when the wait is two cycles).

(5) Self refresh cycle

Although refresh cycle is used only in CBR (auto) refresh cycle, use of CBR self refresh cycle is effective when SDRAM is used for portable equipment, etc. which is driven only with the battery.

Self refresh requires smaller current than CBR (auto) refresh does. For details, refer to individual Data Sheet.

APPENDIX A DIAGRAM AND TIMINGS OF ENTIRE CIRCUIT

Figure A-1 shows the diagram of entire circuit of the circuit examples. Figures A-2 and A-3 show the timings of the logic verification result of this circuit. The timing charts of the logic verification result have taken propagation delay in consideration.

“Name” in Figures A-2 and A-3 show each signal name, and signals shown in parentheses have the following meanings.

[I] : Input signal

[O] : Output signal

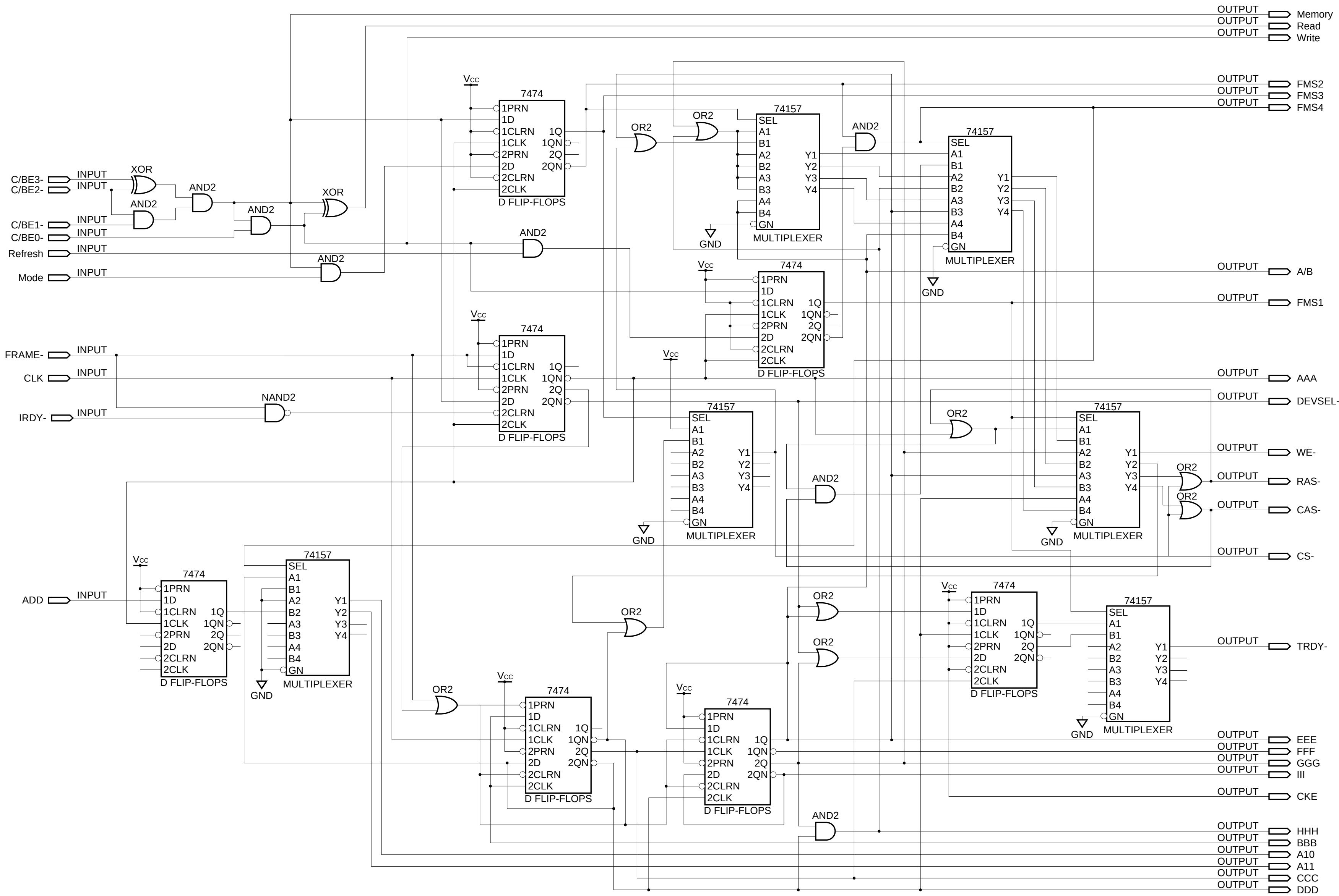
Operating frequency of 66 MHz is used for the logic verification.

In Figures A-2 and A-3, the logic verification is performed with sequence of the following operations.

- Burst read cycle with burst length = 4
- Burst write cycle with burst length = 4
- Burst write cycle with burst length = 4
- Burst read cycle with burst length = 4
- Burst read cycle with burst length = 4
- 1 cycle of CBR (auto) refresh cycle
- 1 cycle of CBR (auto) refresh cycle
- 1 cycle of access cycle other than memory
- 1 cycle of CBR (auto) refresh cycle
- 1 cycle of mode register set cycle

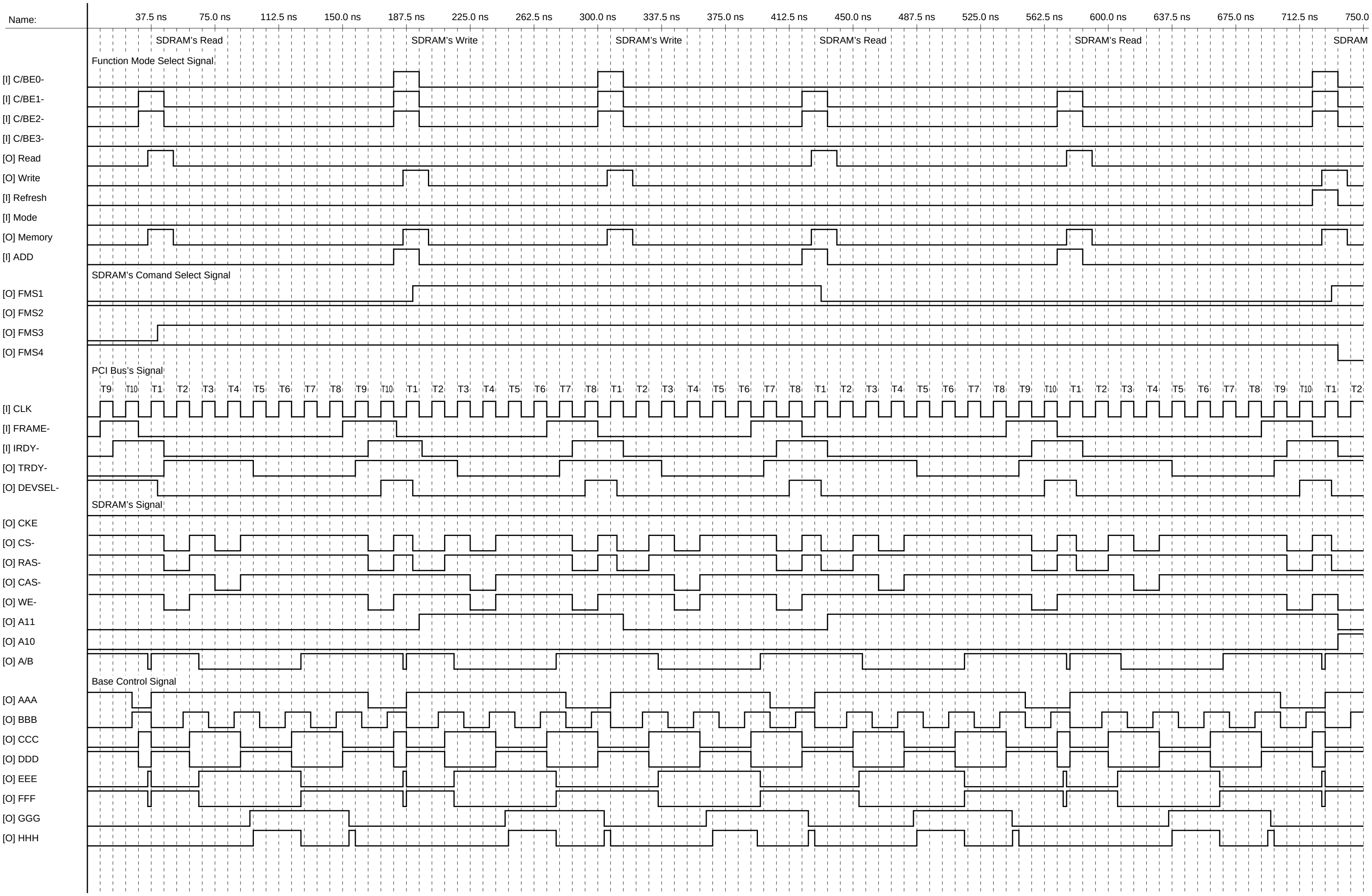
In Figure A-2, 0 ns to 750 ns are shown, and in Figure A-3, 750 ns to 1388 ns are shown.

Figure A-1. Entire Circuit Diagram



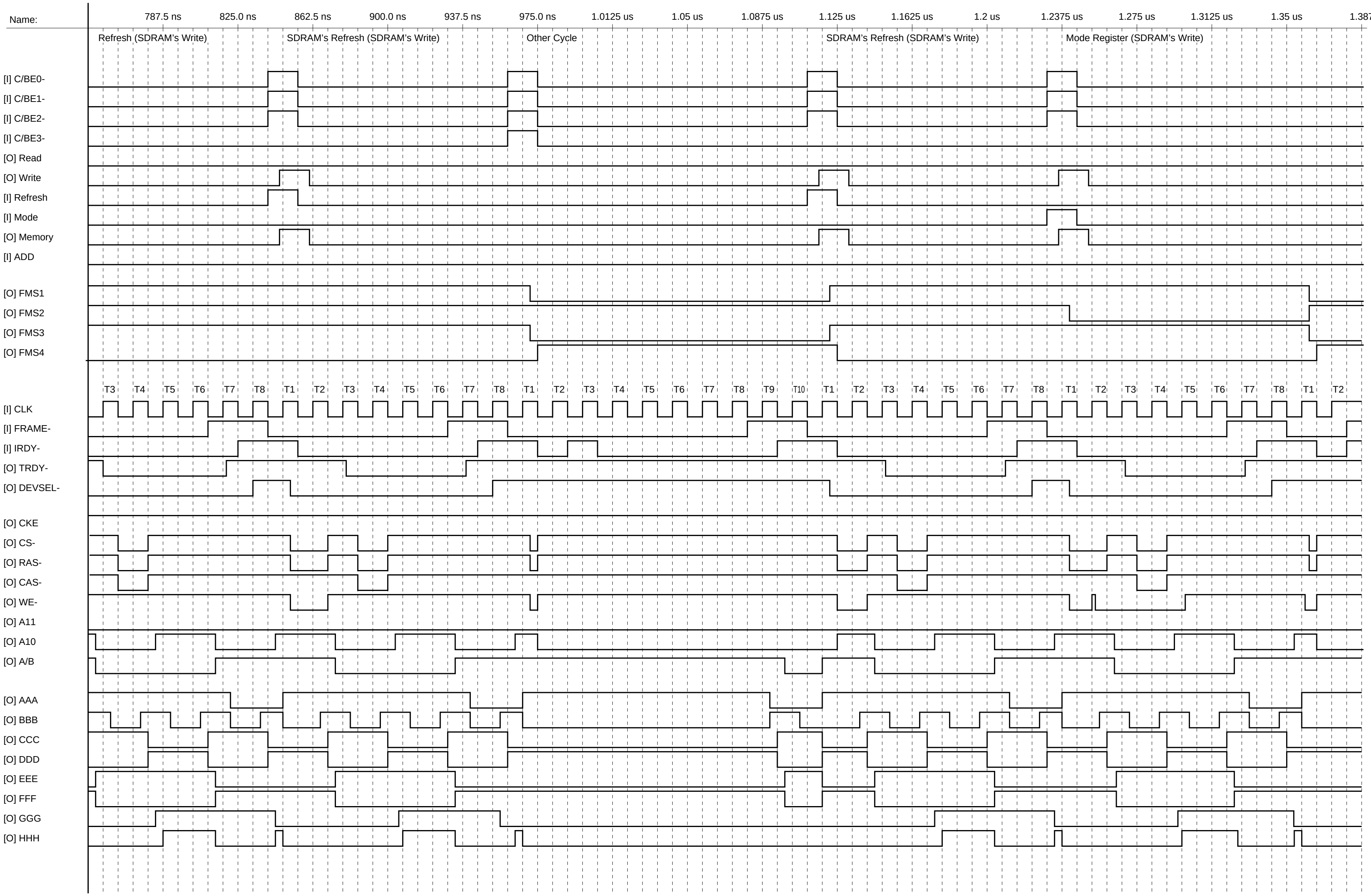
Remark In the diagram, “-” postfixed to pin names and signal names indicates active low.

Figure A-2. Timings of Logic Verification Result (1)



Remark In the diagram, "-" postfixed to pin names and signal names indicates active low.

Figure A-3. Timings of Logic Verification Result (2)



Remark In the diagram, “-” postfixed to pin names and signal names indicates active low.